Masters Program in Geospatial Technologies

UAV - based imagery processing using Structure from Motion and Remote Sensing technology

Ondrej Zvara

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

UNIVERSIDADE NOVA ISEGI
UNIVERSITAT JAUME·I
WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

UAV - based imagery processing using Structure from Motion and Remote Sensing technology

Supervised by

Gabriel Ignacio Guerrero Castex, PhD

Institute of New Imaging Technologies, Universitat Jaume I,

Castellón de la Plana, Spain

Co-supervised by

Torsten Prinz, PhD

Institut für Geoinformatik, Westfälische Wilhelms-Universität,

Münster, Germany

Co-supervised by

Mário Caetano, PhD

Instituto Superior de Estatística e Gestão de Informação,

Universidade Nova de Lisboa,

Lisbon, Portugal

March 2015

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Guerrero for his advices, patience, time and effort he dedicated to my thesis. Next, I would like to thank the co-supervisors, Dr. Prinz for his comments and suggestions as well as to Dr. Caetano for his advices and comments.

I would like to thank to M.Sc. Jan Lehmann for providing me an opportunity to work with his data as well as for the advices, comments and spent time.

Also I would like to express my thankfulness to the European Commission and Erasmus Mundus Master in Geospatial Technologies for providing me the opportunity to study this study program.

Finally I would like to thank to my family, who supported me during my studies and who provided me a stable background, so much necessary for accomplishing this life task.

UAV - based imagery processing using Structure from Motion and Remote Sensing technology

ABSTRACT

With the recent advances in technology and miniaturization of devices such as GPS or IMU, Unmanned Aerial Vehicles became a feasible platform for a Remote Sensing applications. The use of UAVs compared to the conventional aerial platforms provides a set of advantages such as higher spatial resolution of the derived products. UAV - based imagery obtained by a user grade cameras introduces a set of problems which have to be solved, e. g. rotational or angular differences or unknown or insufficiently precise IO and EO camera parameters. In this work, UAV - based imagery of RGB and CIR type was processed using two different workflows based on PhotoScan and VisualSfM software solutions resulting in the DSM and orthophoto products. Feature detection and matching parameters influence on the result quality as well as a processing time was examined and the optimal parameter setup was presented. Products of the both workflows were compared in terms of a quality and a spatial accuracy. Both workflows were compared by presenting the processing times and quality of the results. Finally, the obtained products were used in order to demonstrate vegetation classification. Contribution of the IHS transformations was examined with respect to the classification accuracy.

KEYWORDS

Structure from Motion
Unmanned Aerial Vehicle
Unmanned Aerial System
DSM
Orthophoto
PhotoScan
VisualSfM
IHS

ACRONYMS

API	Application Programming Interface
BBA	Bundle Block Adjustment
BNDVI	Blue Normalized Difference Vegetation Index
CMVS	Clustering views for Multi-View Stereo
CIR	Color Infrared
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DEM	Digital Elevation Model
DGPS	Differential GPS
DSM	Digital Surface Model
EO	External Orientation
GCP	Ground Control Points
GCS	Ground Control Station
GLSL	OpenGL Shading Language
GPS	Global Positioning System
GPU	Graphics Processing Unit
IDW	Inverse Distance Weighted
IHS	Intensity Hue Saturation
IMU	Inertial Measurement Unit
INSPECTED.NET	INvasive SPecies Evaluation, ConTrol & EDucation.NET
IO	Internal Orientation
MSL	Mean Sea Level
NDVI	Normalized Difference Vegetation Index
NIR	Near Infrared
NN1	Nearest Neighbor 1
NN2	Nearest Neighbor 2
OBIA	Object Oriented Image Analysis
OpenCL	Computing Language
OpenGL	Open Graphics Library
PC	Personal Computer

PMVS2	Patch-based Multi-View Stereo
RBMC	Brazilian Network for Continuous Monitoring of the GNSS Systems in real time
RGB	Red Green Blue
RMSE	Root Mean Square Error
S1	Segment 1
S2	Segment 2
SfM	Structure from Motion
SIFT	Scale Invariant Feature Transform
UA	Unmanned Aircraft
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UTM	Universal Transverse Mercator
VTOL	Vertical Take-Off and Landing
WGS84	World Geodetic System 1984

INDEX OF THE TEXT

INDEX OF TABLES

INDEX OF IMAGES

INDEX OF FORMULAS

1 Introduction

Technical development and component miniaturization in the recent years allowed a rapid increase in usage of the Unmanned Aerial Systems (UAS) for various application purposes. In the field of Remote Sensing, UAS proved to be a feasible sensor platforms (WATTS et al., 2012; TURNER et. al, 2012; GUPTA et al., 2013). However, in contrast with conventional aerial imagery, UAS-based data posses a set of specific characteristics such as lack or low precision of the External or Internal parameters, high angular or rotational variability etc. (TURNER et al., 2012; ZHANG et al., 2011). Several algorithms developed in the field of computer vision in the recent years have been applied to process UAS data. In order to extract products from the data serving for further analysis, the camera parameters have to be estimated. A set of problems associated with such a goal was solved by multiple algorithms such as Scale Invariant Feature Transform (TURNER et al., 2012). Structure from Motion is a toolchain providing the solution utilizing various algorithms with purpose to detect features on the images, perform their matching and consequently estimate camera parameters (WESTOBY et al., 2012). Being implemented in various software, users have a set of possibilities how to process their UAS data. However, software solutions can vary in the quality of a result or processing time amount. Moreover, several difficulties in the image processing can raise, such as image blur decreasing the quality or small image overlaps. Such situations can occur due to a platform instability caused by external factors (e. g. wind) or deviations from flight plan caused by insufficient precision of an onboard GPS unit (HUNT et al., 2012; TURNER et al., 2012).

Raising needs of alternative energy resource as well as reduction of CO2 production led to introduction of outlandish plant species in the past century. However, uncontrolled spread of such a vegetation can have an influence on domestic ecosystems. The INvasive SPecies Evaluation, ConTrol & EDucation.NETwork (INSPECTED.NET) project aims to investigate the spread of invasive *Acacia mangium*. A group of specialists in the field of ecology examine the spread of the plant on different scales in Brazil and Portugal (WWU

MUENSTER, 2015). A part of a project proposed by M.Sc. Jan Lehmann (Institute of Landscape Ecology, WWU Muenster, Germany) utilize the Remote Sensing technology. Imagery was collected using UAS technology. In this thesis, data collected by M.Sc. Jan Lehmann within the INSPECTED.NET project will be used in order to present raw image processing workflows and perform results comparison.

1.1 Problem statement and objectives

In order to extract as much information possible from the obtained imagery, raw data have to be processed - images have to be aligned, camera parameters have to be estimated and point clouds generated. Consequently, products such as Digital Surface Model (DSM) and orthophoto can be derived. Such a procedures can be executed using different software and different parameter setup. Finally, quality assessment has to be performed, in order to compare the results quality.

Main objectives of this thesis can be concluded as follows:

- presentation of raw UAS imagery processing workflows using different software
- estimation of optimal software parameter setup for camera parameters estimation within each workflow
- comparison of the resulting products in terms of quality and accuracy
- comparison of workflows feasibility with respect to resulting processing time and quality

Secondary objectives then can be concluded as:

- demonstration of obtained product usage within Acacia classification
- estimation of IHS transformation contribution in terms of classification accuracy

1.2 Text organization

This work is organized as follows. First, theoretical background will be provided in Chapter 2. Study area will be described in Chapter 3. In Chapter 4. different workflows using PhotoScan and VisualSfM software will be presented, resulting in DSM and orthophoto products. Chapter 5. will then demonstrate the usage of obtained data. Finally, Chapter 6. will present the resulting data and workflows comparisons as well as classification results.

2 Background

2.1 UAS in Remote Sensing applications

Unmanned Aerial System (UAS) consists of an aircraft, associated sensors and control equipment. The aircraft is not operated by a human operator, whereas it is controlled remotely or flies autonomously (GUPTA et al., 2013). UAS, also referred as UAVs (Unmanned Aerial Vehicles) have long military tradition. However, in 1990 numerous smaller organizations developed efforts, encouraged by technical improvements and miniaturization of components such as GPS (Global Positioning System) or Inertial Measurement Unit (IMU), in order to modify UAS for the research purposes. Vast variety of UAS is available nowadays for a palette of applications (WATTS et al., 2012; TURNER et. al, 2012). UAS became feasible solution for a range of applications such as search and rescue, real-time surveillance, reconnaissance operations, traffic monitoring, hazardous site inspection or agriculture monitoring (GUPTA et al., 2013).

UAS consists of a two main components: the unmanned aircraft (UA) and Ground Control Station (GCS). According to the different characteristics such as aerodynamics or size, it is possible to classify UA as follows (GUPTA et al., 2013):

- Fixed-wing UA - unmanned airplane with wings, which requires a take-off and landing runway and flies with high cruising speeds and long endurance.
- Rotary-wing UA - also referred as Vertical Take-Off and Landing (VTOL) does not require runway, has ability to hover and flies with high maneuverability. Different hardware configurations (number of motors) are possible (helicopter, coaxial-, tandem- or multi-rotor).
- Blimps - balloons or airships flying at low speed with low maneuverability. Generally blimps have a large size.
- Flapping-wing UA - posses flexible and/or morphing wings inspired by anatomy of birds and insects.

Remote sensing is an established procedure of environmental analysis, performed either using satellite observations or aerial imagery (COLOMINA et al., 2014). However imagery obtained using such a platforms typically reaches spatial resolution between 20 - 50 cm/px. Data with such a resolution are still insufficiently fine for observing e.g. vegetation and its structure (TURNER et al., 2014).

In recent years the use of UAS as a platform for environmental remote sensing had grown rapidly in practical applications as well as in scientific research. Number of studies using imagery obtained by UAS have been conducted with focus on e. g. wildfire mapping, arctic sea ice studies or endangered species detection. In this applications, small user-grade cameras were often used due to the small payload capacity of UAS (LALIBERTE et al., 2011). Using such an approach, e.g. the Mediterranean forests were mapped (DUNFORD et al., 2009), field crops were monitored (HUNT et al., 2010) or mosses in Antarctica were mapped (TURNER et al., 2014).

Due to the lack of high-quality lightweight multispectral sensors suitable for UAS, the alternatives have been developed (LALIBERTE et al., 2011). A typical approach for vegetation detection using UAS is equipping the platform with adjusted camera, where infrared filter is removed and the infrared spectrum is stored in one of the sensors channels (TURNER et al., 2014).

2.2 Traditional Methods

Traditional aerial imagery has several differences, compared to UAV-based imagery. Low flying altitude of UAV provides significantly higher spatial resolution, reaching up to 1 cm/px. High spatial resolution requires significant amount of imagery to be collected in order to cover comparable area as in case of manned aerial flight campaign - covering 2 ha can e. g. require 150 - 200 images (TURNER et al., 2012). UAV imagery often has a large rotational and angular differences and significant perspective distortions due to relatively low flying altitudes. Exterior orientation parameters are usually unknown or inaccurate (TURNER et al., 2012; ZHANG et al., 2011) and due to the usage of a user-grade cameras, the Internal Orientation parameters are unknown or unstable (TURNER

et al., 2012). Spatial resolution, illumination and occlusion are also varying significantly. As those are typical features for terrestrial close range photogrammetry, UAV-based data has characteristics of both terrestrial and conventional aerial imagery (HUNT et al., 2012; BARAZZETTI et al., 2010).

Acquiring fine resolution imagery by the means of satellites or airplanes can get significantly expensive. Therefore UAS proved to provide smaller and affordable platform for remote sensing sensors (COLOMINA et al., 2014).

In order to extract 3D information from 2D imagery, several methods were developed in the field of photogrammetry, in which 3D information for image points occurring on at least two images can be retrieved. These methods, however, require images to have known Internal Orientation (IO) and External Orientation (EO) parameters (VERHOEVEN et al., 2012).

2.3 Structure from Motion

Recent advances in the field of Computer Vision brought new algorithms able to process terrestrial imagery (TURNER et al., 2012). Among them e. g. algorithms as Scale Invariant Feature Transform (SIFT) (LOWE, 1197), which is able to detect features from images or Structure from Motion (SfM) approach able to extract 3D positions of the features derived from imagery (TURNER et al., 2012).

Structure from Motion (ULLMAN, 1979) is a method for high-resolution surface reconstruction. It works on same principals as stereoscopic photogrammetry using overlap of imagery for 3D information extraction (WESTOBY et al., 2012). However, the key difference compared to conventional photogrammetry lies in ability to automatically solve the scene geometry, IO and EO parameters of imagery. These are solved simultaneously by iterative bundle block adjustment method (BBA) performed over a set of features extracted automatically from overlapping images (WESTOBY et al., 2012).

Feature extraction performed by specialized algorithms is based on describing local features in the overlapping images. After their identification, those are matched across multiple images. Using this approach the potential correspondences are identified. Set of correspondences is consequently used for

calculation of their relative positions and thus forming the 3D sparse point cloud representing the geometry. Camera IO and EO are calculated as well in local coordinate system (VERHOEVEN et al., 2012).

The resulting solutions for camera positions are organized in local coordinate system due to lack of spatial information provided by GPS measurements. Thus, the resulting point cloud has to be georeferenced from relative to absolute (space) coordinate system. This can be done using similarity transformation based on provided Ground Control Points (GCP) (WESTOBY et al., 2012).

2.3.1 Scale Invariant Feature Transform

SIFT algorithm can be used as a feature detector in SfM workflow being able to provide high number of features for Bundle Block Adjustment, thus improving the process result (TURNER et al., 2012). Stable points in the scale space are identified by stage filtering, based on identifying areas of local maxima or minima of a Difference-of-Gaussian function. A feature vectors of each point in image is describing the local image region, sampled relatively to its scale space. These vectors are called *SIFT keys*. Matching of the SIFT keys is performed by nearest neighbor approach. If at least three keys are matching with low residuals, the feature object (tie point) is probable to be present. More details about SIFT algorithm can be found in (LOWE, 1999).

2.3.2 Bundle Block Adjustment

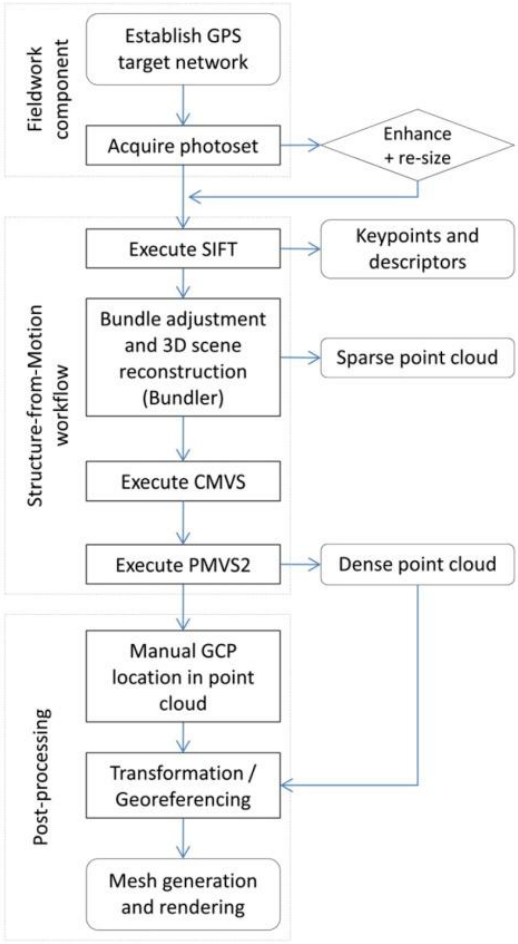
Purpose of the Bundle Block Adjustment system is to estimate the camera parameters and create sparse point cloud (WESTOBY et al., 2012). One of the typical implementations can be found in the open-source project Bundler. Feature objects serve as an input together with a set of images (WESTOBY et al., 2012; SNAVELY et al., 2008). Feature linking tracks are used to reconstruct the point cloud, if the feature is present on at least three images. Features not satisfying this condition are automatically removed. This method ensures non-static objects, such as cars, people or unintentionally captured parts of UAV to be removed from

the reconstruction. However there are features associated with these objects, their relative position compared to the other features is changing constantly (WESTOBY et al., 2012). Positions of the corresponding features create constraints on cameras EO, which is consequently calculated using similarity transformation. Least-square method is used to minimize the estimate of EO. Next, the features 3D position is calculated using triangulation and thus the point cloud is being reconstructed (WESTOBY et al., 2012). More in-detailed description of the procedure can be found in (SNAVELY et al., 2006; SNAVELY et al., 2007).

2.3.3 CMVS and PMVS2

Sparse point cloud produced by bundle adjustment can be densified using Clustering views for Multi-View Stereo (CMVS) and Patch-based Multi-View Stereo (PMVS2) algorithms (WESTOBY et al., 2012). CMVS takes the imagery and the camera positions produced by Bundler as an input and decomposes the images into a clusters of manageable size. It processes the clusters independently, without loss of detailed information (FURUKAWA et al., 2010a). CMVS should be used together with the PMVS2 algorithm which is a multi-view stereo software (FURUKAWA et al., 2010b). It reconstructs 3D structure of the scene from the imagery clusters produced by CMVS and outputs the densified point cloud (WESTOBY et al., 2012). More details about the algorithms can be found in (FURUKAWA et al., 2010a; FURUKAWA et al., 2010b).

As mentioned before, the point clouds (sparse or dense) are generated in local coordinate system, and there is a need to of georeferencing to absolute (spatial) coordinate system. As the accuracy of UAV onboard GPS/IMU are often not sufficient for the direct georeferencing (not relying on GCP) (TURNER et al., 2012), there is a need to perform GCP collection and perform indirect georeferencing in case of high spatial accuracy is required. An example of SfM workflow can be seen on Img. 1.



Img.1.: Example of SfM workflow. Source: WESTOBY et al., 2012.

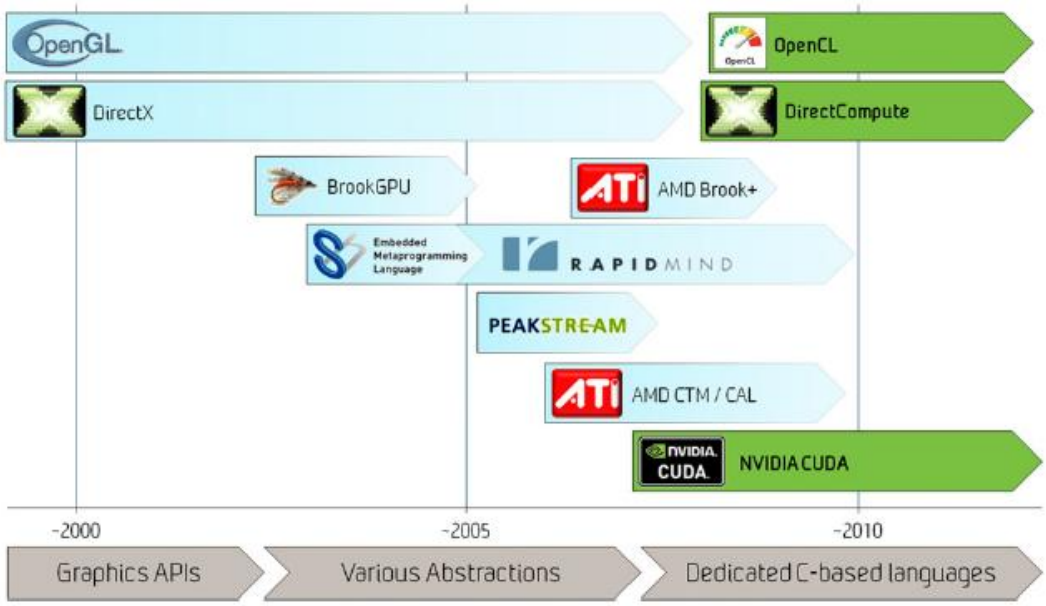
2.4 GPU-based processing

The trend of exponential Central Processing Unit (CPU) performance growth had stopped in the 2000s due to power consumption limitation, which is proportional to squared frequency. Cooling of such a power consumption became a limit, which forced the manufacturers to rise the performance by multiplying number of cores (BRODTKORB et al., 2013).

Meanwhile the exponential growth of Graphics Processing Unit (GPU) performance was recorded due to intensive usage of parallelism concept, as the commonly performed tasks in GPU such as vertex- and pixel-based operations can be performed separately, without dependency on each other (BRODTKORB et al., 2013). Typical features of such an operations can be summarized as follows (OWENS et al., 2008):

- Operations have often large computational requirements.
- High level of parallelism is present.
- Latency of data is less important than throughput.

While CPU distributes its resources in time thus focusing on the data processing in consequent stages, GPU approaches processing by dividing the workflow in stages and in space - the stages are transferring its outputs between each other (OWENS et al., 2008). Operations benefiting from parallel processing can be speeded up by this approach while serial operations can still present a bottleneck. Thus heterogeneous computer concept was developed where applications take advantage of both CPUs serial and GPUs parallel computational power (BRODTKORB et al., 2013). The effort dedicated into development of high-level third-party programming languages able to simplify communication with GPU interface resulted into products such as CUDA, similar to C programming language (OWENS et al., 2008; BRODTKORB et al., 2013).



Img. 2: History of GPU languages development. Source: BRODTKORB et al., 2013.

2.4.1 CUDA

One of the major vendors of high-performance GPUs nowadays is NVIDIA company (BRODTKORB et al., 2013) which developed and released in 2006 its Compute Unified Device Architecture (CUDA) language allowing to forward C, C++ or Fortran code directly to the GPU without a need of any assembly language. CUDA thus presents a programming language as well as computing platform for parallel processing, which has a significant popularity in scientific research (NVIDIA, 2015).

2.4.2 OpenCL

Open Computing Language (OpenCL) is a language tailored for parallel programming able to communicate with CPUs and GPUs of major producers. As it is a non-proprietary language based on public standard, products of companies such as AMD, NVIDIA or INTEL are supporting this language. It is used in a range of applications varying from entertainment to scientific purposes (SCARPINO, 2011).

2.4.3 OpenGL and GLSL

Open Graphics Library (OpenGL) is an Application Programming Interface (API) introduced by Silicon Graphics Computer Systems in 1994. This library supports wide range of rendering, texture, mapping etc. effects. Similarly to previously mentioned OpenCL, OpenGL is also cross-platform library independent on operating system or hardware configuration. The structures used in OpenCL-based programs are formed by 3D objects built from geometric primitives such as points, lines, triangles and patches (SHREINER et al., 2013; OPENGL, 2015a).

OpenGL Shading Language (GLSL) is a part of OpenGL since version 2.0 Its purpose is to enable expressing of complex processing operations such as vertex or fragment processing. Main features of the language are independently compiled units - shaders, which after being linked together form the GPU program. (OPENGL, 2015b).

2.5 Software implementations of SfM

Currently SfM is being implemented by several softwares, providing different forms of processing approach. Generally, softwares can be grouped as an individual algorithms (SIFT), packed solutions (PhotoScan, VisualSfM, Pix4D) or web services (Photosynth, 123D Catch) (GREEN et al., 2014; TORRES et al., 2012).

2.5.1 PhotoScan

PhotoScan, a product of Russian company AgiSoft LLC., is a commercial software most directly comparable to the Bundler open-source software and the PMVS2 (GREEN et al., 2014). PhotoScan approaches image processing in a similar manner as the SfM approach. GUI guides user through the steps of processing which is thus becoming more convenient. During the workflow, PhotoScan allows user to specify several parameters influencing processing time and the result quality (VERHOEVEN, 2011).

Typical workflow in PhotoScan is divided into a few steps. First, imagery is loaded, then feature extraction, matching and bundle adjustment are performed resulting in a sparse point cloud, IO and EO camera parameters estimates (GREEN et al., 2014). After these steps are executed sparse point cloud can be georeferenced using GCPs. The GCPs can also serve for camera IO and EO optimization (AGISOFT LLC, 2015). Next, the dense point cloud is computed and consequently mesh can be approximated (GREEN et al., 2014). Having these products created, orthoimagery and DSM can be exported (AGISOFT LLC, 2015).

The exact modifications of algorithms that are implemented in Agisoft PhotoScan are not known (GREEN et al, 2014).

PhotoScan support GPU computing utilizing OpenCL enabled GPU devices. The GPU acceleration is used to speed up the process of point cloud densification. The availability of GPU-based processing depends on the platform, software driver versions and video chip types (AGISOFT LLC, 2015).

Recommended software configuration for version 1.0.0 is 64bit operating system (Windows XP or higher / Mac OS 10.6 or higher / Linux Debian/Ubuntu), decent multicore CPU (Intel Core i7) and 12 GB RAM. The amount of possibly processed photos as well as the parameter setting used during the processing are dependent on amount of RAM available (AGISOFT LLC, 2015).

2.5.2 VisualSfM

Open-source SfM workflow for obtaining results such as 3D model utilizes various different software such as Bundler or CMVS/PMVS2. Performing steps in SfM using these software separately one by one can be time consuming and inconvenient, especially compared to some of the commercial implementations (GREEN et al., 2014; TORRES et al., 2012).

One of the software options providing packed open-source solution is presented by VisualSfM (WU, 2013; WU et al., 2011). The software incorporates SfM chain using Graphical User Interface (GUI) which makes usage of the algorithms easier (GREEN et al., 2014). The algorithms leading to sparse point cloud reconstruction are implemented using multi-core parallelism, thus the computations are faster compared to standard implementations of respective algorithms (GREEN et al, 2014). Dense reconstruction is then performed using CMVS/PMVS2 package (WU, 2015f).

The software GUI allows user to load images and by simple steps perform the feature detection and matching. Consequently BBA is performed and the sparse point cloud can be visualized in 3D viewer. PMVS2 package then performs the cloud densification resulting in a series of .PLY format point clouds. Those can be visualized using the 3D viewer, however the point clouds can not be edited, thus external software has to be used (GREEN et al., 2014). CMVS/PMVS2 software does not come along with the VisualSfM and its binaries have to be downloaded and placed in the file structure manually (WU, 2015e). The implementations of algorithms supporting GPU computation used for sparse cloud reconstruction are described below.

2.5.2.1 **SiftGPU**

SiftGPU algorithm implements SIFT (LOWE, 1999) for processing using GPU. Steps such as image up- or down-sampling, Gaussian pyramids or Difference-of-Gaussian building, key-point detection, feature list generation or feature orientation description can be performed using multi-core computation (WU, 2015a). Using multi-core computation the processes can be executed significantly faster compared to standard SIFT implementation. As not all of the steps can be performed faster compared to standard SIFT, SiftGPU also tries to find the optimal settings for each step (WU, 2015a). SiftGPU thus performs operations using both CPU and GPU (WU, 2015a). The algorithm also supports feature matching using GPU. Currently it supports both CUDA and GLSL languages (WU, 2015b).

2.5.2.2 **Preemptive matching**

In order to calculate relative positions of the cameras, feature matching has to be performed. Matching is an iterative process, where the results of each iteration are checked whether the constraints are satisfied. If the detected match does not satisfy positioning constraint established in previous iterations, it is discarded. Thus with increase of the image amount, the time of matching increases exponentially (GREEN et al, 2014). In order to reduce the processing time which has a significant contribution to the overall SfM processing time, matching can be divided into more cores or computers. As the imagery of large datasets introduce significant amount of viewpoints variety, the majority of the candidate matches does match. Thus, by identifying good pairs robustly, significant amount of computation time can be saved (WU, 2013). VisualSfM comes with option of the *preemptive matching* approach based on the features sorting with respect to their scale in the feature scale space (WU, 2013). Features of each image are organized in decreasing scale order and only a subset of those is matched with a pair candidate image features organized in the same manner. If the number of matches exceeds the given threshold, the full image pair matching is performed. Otherwise the image pair is skipped. Top-scale features are suitable

for the matching as they are likely to be present also in lower image scales of Gaussian levels (WU, 2013).

2.5.2.3 **Multicore Bundle Adjustment**

Multicore Bundle Adjustment is an implementation of Bundle Adjustment algorithm utilizing both CPU and GPU cores for 3D model reconstructions of large image datasets. Parallel distribution of the processing workload provides significant time as well as memory savings (WU, 2015c).

Processing using VisualSfM requires decent GPU. The exact specifications are not provided however reasonable amount of GPU memory is required - 1GB of GPU at least. In particular, small GPU memory can cause problems while detecting the features (WU, 2015d).

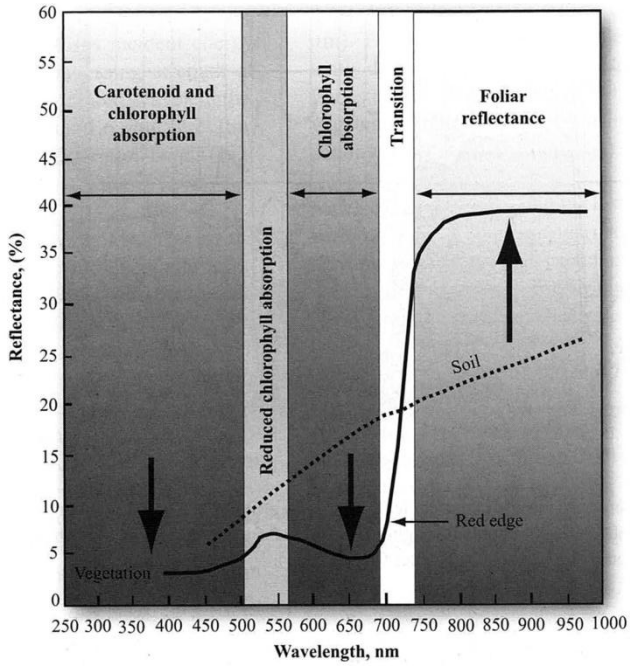
2.6 Image enhancements

2.6.1 Vegetation indices

Extraction of vegetation biophysical properties such as chlorophyll content, percentage of cover or amount of biomass can be done using vegetation indices (JENSEN, 2005). Vegetation indices are arithmetic calculations over bands of multispectral image with purpose to enhance the vegetation content of the image (DOBROVOLNY, 1998). These calculations should maximize the biophysical element enhancement, eliminate external or internal effects such as illumination differences of scene or topography and should relate to the certain biophysical parameter, e. g. biomass amount (JENSEN, 2005).

The concept of vegetation indices is based on the knowledge of vegetations spectral behavior. Energy reaching earth surface can be either reflected, absorbed or transmitted dependent on the surface character and energy wavelength. In case of vegetation, majority of NIR energy is either transmitted or reflected while only a small amount is absorbed. On contrary wavelengths of visible electromagnetic spectrum are mostly absorbed and a small amount is

transmitted or reflected (JACKSON et al., 1991). Spectral behavior of vegetation can be seen on Img. 3.



Img. 3.: Spectral behavior of vegetation. Source: JENSEN, 2005.

Normalized Difference Vegetation Index (NDVI) is a ratio of NIR and visible energy. It can be calculated using following formula (NASA, 2015):

$$NDVI = (NIR - VIS) / (NIR + VIS) \quad (1)$$

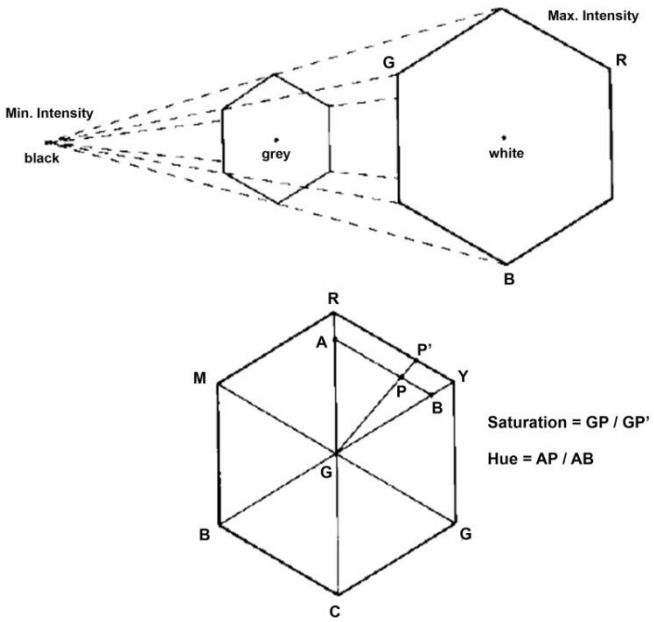
NDVI is an important index due to its feasibility for vegetation seasonal changes monitoring or ability to reduce noise sources such as illumination, topography variability or shadows presence (JENSEN, 2005).

2.6.2 IHS transformation

In order to interpret reliably the information provided by remote sensing imagery, a good presentation of the data is essential. Color composites used for presentation of the data are formed by assigning channels of the multispectral

image to RGB channels of color monitor (DUTRA et al., 1988). There are several techniques developed for better data color enhancements. One of them is IHS transformation, which is based on representing vectors of primary colors (Red, Green, Blue) by three independent attributes - Intensity, Hue and Saturation. Intensity is the amount of black and white, expressing the amount of energy reflected by the surface. Hue represents the amount of color and thus is expressing the dominant wavelength (DOBROVOLNY, 1998). Saturation is expressing the amount of grey in the natural color (DUTRA et al., 1988).

One of the methods to transform RGB color space to IHS space is by projecting RGB color cube to the plane, which is perpendicular to the cubes diagonal representing the amount of grey. Resulting hexagons cusps are representing RGB colors and supplementary Cyan, Magenta and Yellow colors. Position of the projection plane on the diagonal defines the Intensity, while Hue is defined by the angle on the hexagon and Saturation is defined as distance from the center of the hexagon (DOBROVOLNY, 1998). Explanation and definition of Saturation and Hue calculation can be seen on Img. 4.



Img. 4.: Principles of IHS transformation. Source: LILLESAND et al., 1994; DOBROVOLNY, 1998.

User grade cameras typically used as a sensors due to payload capacity of the UAS platforms often provide limited spectral resolution. An option how to compensate lack of spectral and radiometric resolution of such a kind of imagery is to use Intensity, Hue and saturation information of the imagery (LALIBERTE et al., 2008; LALIBERTE et al., 2011). Inter-correlation between RGB bands is relatively high, while opposite situation is observed within IHS bands (LALIBERTE et al., 2007).

IHS transformation proved to be useful in analysis of RGB terrestrial imagery where e. g. Intensity band served as a tool for bare ground detection using thresholding (LALIBERTE et al., 2011). Laliberte et al. (2007) used Intensity band for shadow masking, which is often present in high-resolution imagery. Saturation and Hue bands were moreover used in the same study for vegetation and soil distinction, which allowed more precise classification between senescent and healthy vegetation. Next, IHS transformation was used by Karcher et al. (2003) for turf grass cover analysis or by Koutsias et al. (2000) for burned land mapping.

2.7 Surface elevation extraction

The height difference between Mean Sea Level (MSL) and the bare earth surface is referred as elevation. Digital representation of elevation is called Digital Elevation Model (DEM) (BANDARA et al., 2011). Digital Surface Model (DSM) represents surface of earth, including all objects existing on the surface, such as cars, trees and houses (PCI GEOMATICS, 2015). One of the issues associated with DEM extraction is the nature of known automated techniques for image matching which are percepting energy reflected from the top of the surface thus creating a DSM. If there is a need of DTM product, a technique for its extraction from DSM needs to utilized (BANDARA et al., 2011).

One of the solutions of this problem is provided by commercial software PCI Geomatica. The DSM2DTM algorithm performs series of filtering steps over the DSM, and thus it is removing undesired features. After the filtration, it performs smoothing using median filter. The process is controlled by user, and is based on try-error approach, when user has to determine the best parameters of the algorithm based on the empirical experience (PCI GEOMATICS, 2014).

2.8 Object Oriented Image Analysis

Pixel based image classification performed over high-resolution imagery often suffers from inability to extract maximum desired information. Such a situation can occur in the case of spectrally complex areas of urban land cover, where per-pixel classification experiences specific limitations in human-made and natural surface distinction. Approach of the classification based on the spectral and spatial information is referred as Object Oriented Image Segmentation (JENSEN, 2005).

Basic building blocks of the Object Oriented Image Analysis (OBIA) are image segments (objects) - continuous homogenous areas of image (HOFMANN, 2001). Image objects provide much more descriptive power compared to standalone information contained in each pixel. Thus, it is possible to classify image using contextual information such as neighborhood relationships and information obtained by multiple scale objects (objects of different size) (TRIMBLE, 2011a).

One of the software able to perform OBIA is Definiens eCognition. Its multi-resolution segmentation algorithm is able to segment the image into objects and to organize them in hierarchical way (TRIMBLE, 2011a).

Typical workflow of OBIA in eCognition involves two steps. In the first step, so called segmentation, the image is cut into segments. This task can be performed using various algorithms. Next step is the classes assignment to these objects, based on objects attributes such as shape, color, position etc. Iterations of these two steps can be performed as many times as needed, dependent on desired detail level of the classification (TRIMBLE, 2011a).

In eCognition there are multiples algorithms available for segmentation execution (TRIMBLE, 2011b).

2.8.1 Multiresolution segmentation

Multiresolution segmentation algorithm segments the image by minimizing the heterogeneity of objects (TRIMBLE, 2011b). It can perform the segmentation on the pixel level or on the already existing object level (domain). The principle of segmentation is based on pairwise region merging, thus it

consecutively merges pixels (or objects) while minimizing the internal objects heterogeneity and maximizing their homogeneity.

The process starts with single-pixel objects seeded on the image. These are repeatedly merged in a loop if they satisfy the constraint of the local homogeneity threshold. The seeds are searching for the most suitable neighbor candidate. If the suitability is not mutual, the candidate object becomes next seed object and searches for his most suitable neighbor. If the objects are suitable, they are merged. In every loop each object is manipulated only once. The iterations are terminated if there are no other objects to be merged (TRIMBLE, 2011b). The homogeneity criterion is formed by combination of shape and spectral homogeneity and its calculation can be controlled by the user, adjusting the scale parameter of the algorithm. Higher values of the parameter will allow to create larger objects and vice-versa. The advantage of this algorithm lies in its high quality of the results, the drawback is its slower performance. (TRIMBLE, 2011b). Parameters of multiresolution segmentation are listed below:

- *Layer weight* specifies importance of the information contained in the band. The higher the weight, the more important will be the value of the used pixel (TRIMBLE, 2011b).
- *Scale* controls the resulting size of the objects. This parameters determines the maximum allowed heterogeneity inside of an object, thus the size of the resulting objects depends also on the heterogeneity of the input data (TRIMBLE, 2011b).
- *Homogeneity* is composed by the three criterions: color, smoothness and compactness (TRIMBLE, 2011b).
- *Shape* adjusts the sensitivity to of the resulting objects towards the shape, while affecting the weight of the color parameter. By rising the shape parameter, the objects are created with respect to their shape, while being less formed based on their spectral characteristics (TRIMBLE, 2011b).
- *Compactness* optimizes the compactness of the objects. This parameters is useful when separation of objects depends on relatively weak spectral difference (TRIMBLE, 2011b).

2.8.2 Classification in eCognition

There are several ways how to classify image objects in eCognition. The simplest option is to use an *Assign Class* algorithm. This algorithm is using threshold condition to evaluate membership of an object to a class. Another option is *Nearest Neighbor Classification* algorithm. It uses class descriptions to evaluate the membership of the objects. Classes are described by samples trained by user. Samples are image objects that are significant representatives of respective classes. The procedure consists of two steps. Firstly samples (training areas) for each class are provided by user. Next the algorithm classifies objects with respect to their nearest sample neighbor - it searches nearest sample in the feature space of the examined object (TRIMBLE, 2011a).

2.8.3 OBIA and UAV data

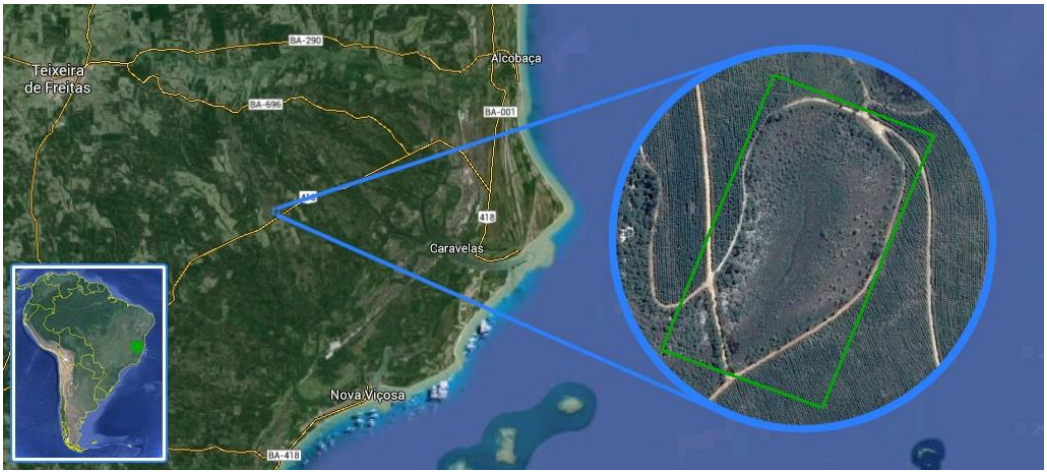
While working with high-resolution imagery, object-based image classifications is performing successfully or even better than pixel-based classification. High-resolution imagery often contains significant amount of shadows, low spectral information or high amount of noise (YU et al., 2006; HODGSON et al., 2003; LALIBERTE et al., 2009). Another reason for OBIA usage suitability with this type of data is caused by high spatial variability between meaningful objects on the image (LALIBERTE et al., 2011). This method has been successfully applied on high-resolution satellite data (SHIBA et al., 2006, D'OLEIRE-OLTMANNS et al., 2014). UAV imagery was successfully processed using OBIA as well (LALIBERTE et al., 2011). In ecology, OBIA allows to detect landscape patches enabling to understand the present processes (LALIBERTE et al., 2007; LALIBERTE et al., 2011).

3 Resources

3.1 Study area

Study area of this work is situated in municipality of Caravelas, south of Bahia State, Brazil. Mussununga, an ecosystem associated with Atlantic Rainforests of southern Bahia and northern Espirito Santo states located in Brazil lies in the evergreen vegetation zone. This ecosystem hosts typical vegetation forms ranging from grasslands up to woodlands (SAPORETTI-JUNIOR et al., 2012).

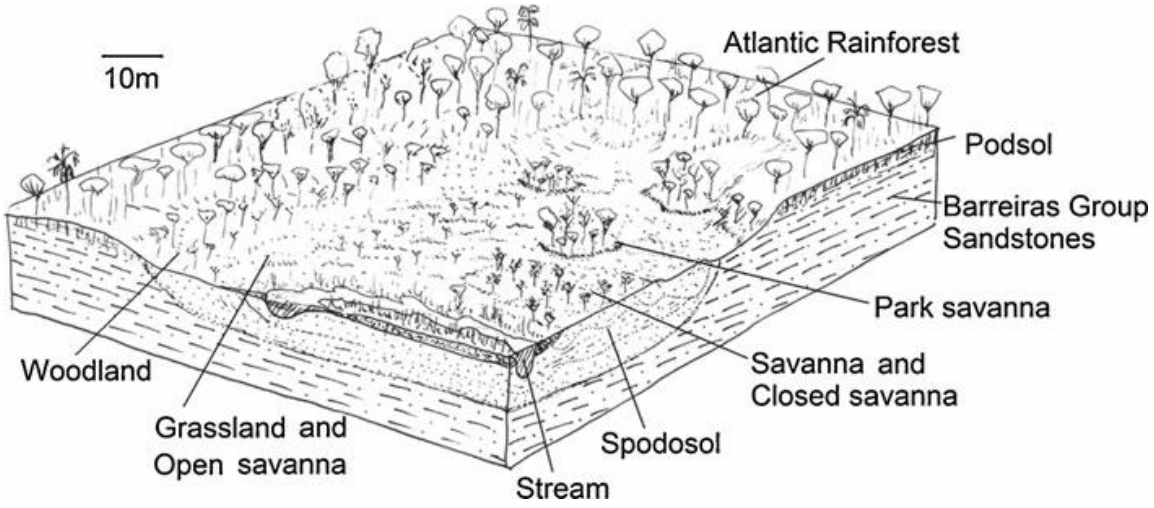
Climate of the area is of Tropical monsoon type, classified as Am level in Köppens classification system. Thus, it is characterized as humid climate, with yearly mean precipitations of 1750 mm. Humid summers are changed by moderate dry winters. (SAPORETTI-JUNIOR et al., 2012).



Img. 5.: Location of study area.

Main vegetation type of this area is presented by rainforests, which are surrounding Mussununga areas consisting mainly of savannic vegetation. Typical are Eucalyptus plantations surrounding the areas of Mussununga (SAPORETTI-JUNIOR et al., 2012).

The typical soils are represented by Spodosols, sandy soils created during Tertiary period (approx 16. mil. years before). The terrain is flat, formed by Tertiary sandstones of Barreiras Group (SAPORETTI-JUNIOR et al., 2012).



Img. 6.: Mussununga vegetation structure. Source: SAPORETTI-JUNIOR et al., 2012.

In this work, two subsets of the Caravelas Mussununga area were selected. Both subsets (segments) have area 2500m^2 (50 x 50m) and capture main vegetation types present in the ecosystem. In this text, the segments will be referenced with abbreviations S1 and S2 respectively. Locations of the segments with respect to the study area can be seen on Img. 7.

3.2 Materials

All materials as well as the field work was done and provided by M.Sc. Jan Rudolf Karl Lehmann as part of his research project mentioned in Chapter 1.

3.2.1 Platform

As a platform for the imagery acquisition an adjusted fixed-wing UAS based on Skywalker model fuselage using 1900mm wings was used. This platform was chosen due to strong wind presence in the area, when fixed-wing platform is able to cope better with such a conditions, compared to copter-type UAS. Next, the long flight time of fixed-wing UAS is essential in this kind of application, as large areas of Mussununga has to be covered. Due to size of the UAS, it is easier to fly in high altitudes.

3.2.2 Imagery and cameras

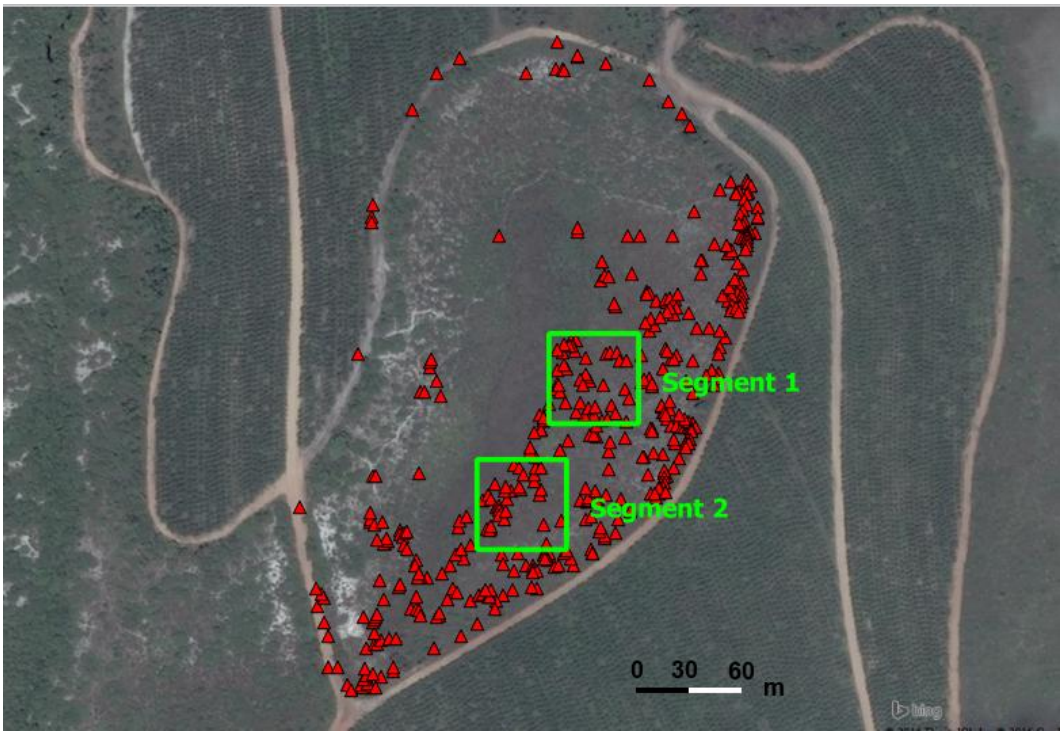
Imagery was obtained in different flight campaigns, using cameras able to capture RGB and NIR wavelengths. First campaign was performed by research team on 16/09/2014 using Canon PowerShot SX230 HS. 207 RGB images were collected. Second campaign was conducted by the team on 25/06/2014 using modified Canon IXUS 100 IS. The so called 'hot filter' filtering NIR in most commercially available cameras was removed. Due to this modification, camera sensor was able to capture NIR wavelengths. The advantage of this modification consists in higher flexibility of image acquisition regimes, as user can decide in which mode the images will be taken by placing respective external filters in front of the camera (LEHMANN et al., 2015). As cameras usually do not have specified channel for NIR wavelengths, these have to be stored in one of the three channels of the sensor. Thus the camera used in this work had attached Cyan color filter to exclude the visible red light and allow the NIR radiation of up to 830nm to be recorded in Red band of the sensor. Camera captured visible and NIR wavelengths simultaneously. The Red light wavelengths were not captured anymore, thus the resulting false color composites were composed by captured *NIR*, *Green* and *Blue* bands. Similar approach was used by Hunt et al. (2010). More details about this technology can be found in (LEHMANN et al., 2015). 185 CIR images were collected using this camera during the second flight campaign. Details of the imagery can be found in Tab. 1.

type	RGB	CIR
date	16/09/2014	25/06/2014
camera model	Canon PowerShot SX230 HS	Canon IXUS 100 IS
number of images	207	185
width x height [px]	3000x4000	3000x4000
avg. size [MB]	2.09	3.00
exposure [s]	1/1600	1/640
aperture	f3.1	f3.2
Focal length [mm]	5	6
GPS data	no	no

Tab. 1.: Detailed information of acquired imagery.

3.2.3 GPS data

There were two types of GPS data collected by the team. Firstly, 15 Ground Control Points (GCPs) were collected using GPS receiver with vertical accuracy of 8.5 cm and horizontal accuracy of 5.7 cm. Next, Differential GPS measurements of Acacia trees in the area were collected. 438 records were captured in the whole area. The overview of data and segments location can be seen on Img. 7.



Img. 7.: DGPS Acacia measurements and location of segments.

4 Implementation

Workflow leading to the extraction of the products such as orthoimagery and digital surface elevation model suitable for further processing involves two main stages. First stage consists of a fieldwork - photo acquisition and GPS measurements (GCPs and DGPS acacia truthing collection). Second stage is then the processing of raw images and GPS data. DGPS measurements have to be corrected. Images of low quality have to be sorted in order to avoid false matches. Image matching is then performed using different software - PhotoScan and VisualSfM. Both workflows fundamentally present similar approach, however due to software capabilities there are some differences in the order of the performed steps - namely, georeferencing part in case of PhotoScan can be done immediately after sparse cloud reconstruction.

The processing steps were done using high-performance PC with NVIDIA Tesla C1060 GPU (4GB GPU memory, 240 cores).

4.1 Data preparation

4.1.1 Differential GPS correction

Acacia ground truthing DGPS records had to be corrected in order to reach higher accuracy. Corrected data were consequently used as GCPs as well as for visual detection of the Acacias on the imagery. The DGPS correction was performed using *GTR Processor* software package. Images were corrected using Brazilian Network for Continuous Monitoring of the GNSS Systems in real time (RBMC), real-time positioning service for DGPS measurements (IBGE, 2015). Finally, 438 records were corrected utilizing *Teixeira de Freitas (BATF)* reference station and exported into WGS84 / UTM 24 S coordinate system.

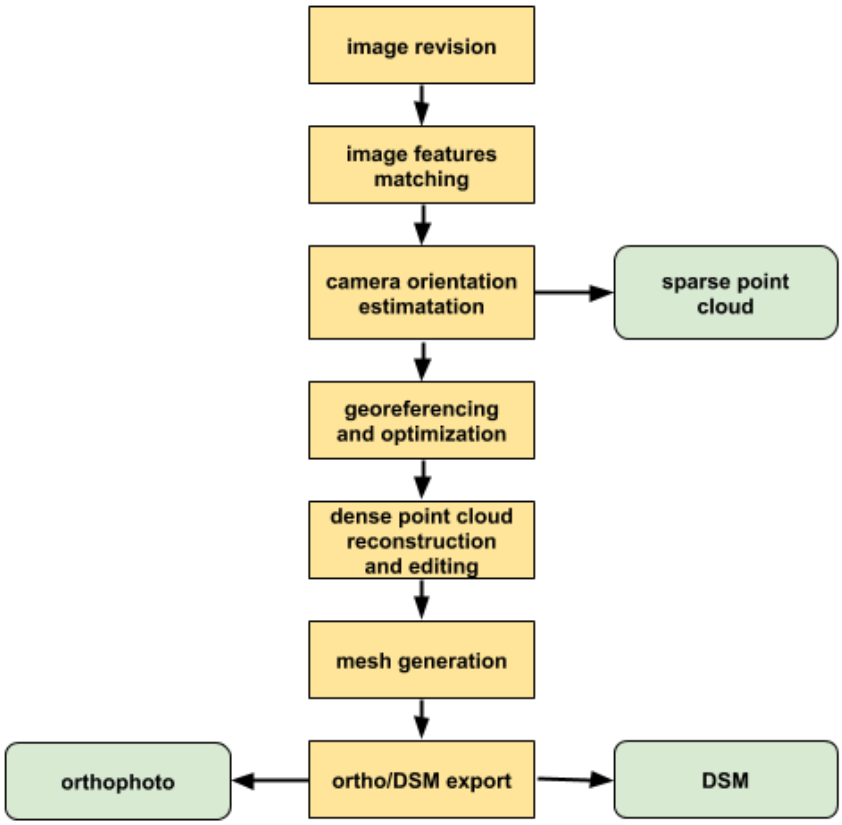
4.1.2 Image selection

Prior to further processing steps, imagery of low quality from both datasets had to be excluded. There are various sources of quality degradation such as platform instability or deviation from flight plan path due to inaccuracies of the

UAS GPS module. The blurred images, high angle oblique images and images not covering the area of interest were filtered resulting in dataset of 195 RGB images and 143 CIR images.

4.2 Image processing using Agisoft PhotoScan

Agisoft PhotoScan software provides a compact commercial solution for the raw imagery processing. In order to demonstrate the workflow, PhotoScan Professional Edition 1.0.4 build 1847 (64 bit) version was used. The typical workflow is divided into several steps. Firstly images are loaded, then feature detection, matching and bundle adjustment steps can be performed resulting in the sparse point cloud geometry reconstruction. After the sparse point cloud is created, the indirect georeferencing and camera alignment optimization can be performed. Camera optimization process provides accuracy enhancement of the camera EO and IO estimates based on the provided GCPs. GCP markers have to be placed manually either using imagery or mesh created based on the sparse point cloud. After the optimization and georeferencing the dense point cloud can be reconstructed. Dense point cloud has to be revised and false matches have to be removed in order to avoid anomalies such as mesh holes etc. Consequently mesh can be created, which allows the export of orthophoto and/or DSM. Additionally, textured mesh can be derived however this step is not necessary in order to export the orthoimagery. Diagram of PhotoScan-based workflow can be seen on Img. 8.



Img. 8.: Workflow diagram of data pre-processing using PhotoScan.

4.2.1 Feature detection and matching

After loading the images into PhotoScan, the first step is to align the images. This procedure is based on the feature detection and matching allowing to estimate positions of the cameras. Thus, this step is important, as point cloud densification can perform good only if sufficient amount of the cameras have known IO and EO parameters. The three involved processes are implemented in PhotoScan as a single step. PhotoScan provides three parameters influencing the final result of the image stitching. Meaning of the parameters is presented below:

- *Accuracy* - controls the accuracy of camera position estimates. Lower accuracy results in lower precision and shorter computation time (AGISOFT LLC., 2014).
- *Pair selection* - controls the manner of image selection for feature matching. *Ground Control pre-selection* option chooses images based on

their position. *Generic* option chooses overlapping images by matching them firstly using low accuracy (AGISOFT LLC., 2014).

- *Point limit* - upper limit of the feature points detectable in the image (AGISOFT LLC., 2014).

In order to estimate a parameter setup with the best performance, feature detection, matching and bundle adjustment were performed using various combinations of the parameters. As mentioned before, number of images with successfully estimated IO and EO parameters is important in order to perform further steps, thus this is the main criterion for the assessment of the process quality. In terminology of PhotoScan, images with known IO and EO estimates are called *"aligned"*.

Images of both datasets were matched using *Generic* method. In both datasets matching was tested using different values of *Accuracy* parameter (low/medium/high) and different values of *Point limit* parameter (40000/80000/120000). Consequently the best matching results were taken for further processing. The results of stitching using different parameters can be seen in Tab. 2. and Tab. 3.

accuracy / point limit	feature detection		feature matching		BBA		
	avg. features detected/img	time [s]	matches	time [s]	sparse cloud points	aligned photos	time
low / 40000	5473	178.324	1005761	466.25	110666	195	231.839
low / 80000	5473	129.547	1005548	446.707	109997	195	194.835
low / 120000	5473	135.269	1005422	450.177	110674	195	191.551
medium / 40000	23378	245.513	1908976	938.202	306565	195	514.442
medium / 80000	23379	240.35	1890303	889.498	306241	195	452.697
medium / 120000	23413	226.813	1862238	856.063	306194	195	455.82
high / 40000	39535	472.04	2104630	1360.033	371496	195	525.557
high / 80000	75626	622.113	2399683	2694.53	461373	195	794.915
high / 120000	86549	643.121	2452558	3191.608	477678	195	646.956

Tab. 2.: Results of RGB imagery stitching using different variations of parameters using PhotoScan.

accuracy / point limit	featrue detection		feature matching		BBA		
	avg. features detected/img	time [s]	matches	time [s]	sparse cloud points	aligned photos	time [s]
low / 40000	5621	99.778	175581	216.138	17618	35	33.057
low / 80000	5621	101.79	175889	215.655	27828	62	32.745
low / 120000	5621	99.783	175939	216.376	17528	35	33.21
medium / 40000	26743	180.211	362748	311.86	54593	35	63.866
medium / 80000	26743	186.29	362666	337.874	54702	35	82.431
medium / 120000	26743	184.545	362662	328.77	54591	35	86.587
high / 40000	39972	348.567	399670	390.281	63469	38	71.105
high / 80000	78011	494.318	436569	799.127	73744	37	92.04
high / 120000	84862	461.015	437898	733.281	73744	38	127.166

Tab. 3.: Results of CIR imagery stitching using different variations of parameters using PhotoScan.

The decision of the final parameters chosen for the stitching was based on Tab. 2., Tab. 3. and visual assessment of the camera positions estimation quality. For the RGB dataset the parameters setup of *Accuracy*: high; *Pair selection*: Generic; *Point limit*: 80000 was chosen. Option using *Point limit*: 120000 parameter value returns similar performance, however processing time is higher. In case of CIR dataset, the setup of *Accuracy*: low; *Pair selection*: Generic; *Point limit*: 80000 was chosen as a suitable configuration due to nearly doubled amount of aligned imagery compared to other cases.

After the alignment was performed, PhotoScan provides an option to reset alignment of incorrectly aligned imagery and perform the procedure again for those. Another option is to introduce the tie points manually, however in this work manual intervention is avoided due to the aim of achieving the automated processes workflow.

4.2.2 Georeferencing and camera optimization

Georeferencing was done using provided DGPS measurements. In order to perform comparison of the results later in this work, subset of DGPS measurements was used, as the different workflows and datasets do not provide exactly the same extend of area covered. Thus a set of Acacia measurements were used in the both datasets. However, it was not possible to use exactly the same points in both cases. Both datasets were georeferenced in WGS84 / UTM 24 S projection. In case of the RGB dataset, 14 GCPs were used. While working with CIR dataset 10 GCPs were placed successfully. Consequently, based on provided GCPs the IO, EO parameters and point coordinate estimates were optimized.

4.2.3 Point cloud densification

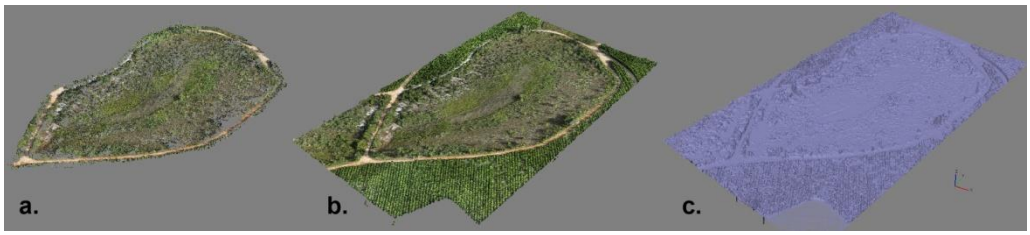
Next step in the workflow was the point cloud densification. This step creates dense point cloud based on the sparse cloud reconstructed in previous step. Agisoft offers two parameters to control this process: *Quality* and *Depth Filtering Mode*. While *Quality* parameter provides control over the quality of resulting cloud (higher quality brings higher computation time), *Depth Filtering Mode* controls outliers filtering (AGISOFT LLC., 2014). In case of the both datasets the parameters were set according to the software recommendations and time processing constraints as follows: *Quality*: *Medium*, *Depth Filtering Mode*: *Aggressive*. The occurring outliers were removed by manual editing.

4.2.4 Mesh reconstruction

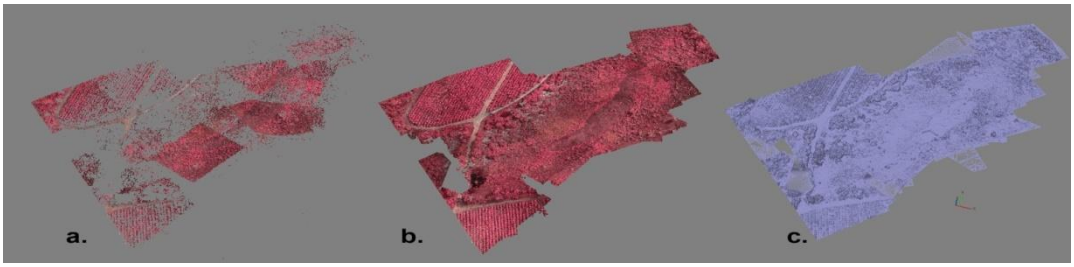
In the PhotoScan mesh can be reconstructed either using interpolation or simply relying on the data created by previous steps. If *Interpolation* parameter is disabled the data are reconstructed based on the input without any other estimates, thus holes can occur in the final mesh. *Source data* parameter can be set to dense or sparse point cloud. *Polygon count* parameter limits the amount of faces. *Surface type* parameter specifies the method for modeling. Dependent on the value, it assumes the types of objects to be modeled (AGISOFT LLC., 2014).

With respect to the software recommendations the parameters for both datasets were set to: *Interpolation*: enabled, *Source data*: dense point cloud, *Polygon count*: high, *Surface type*: height field.

After the mesh reconstruction the DSM and orthophoto export is enabled. For the RGB dataset, both of these were exported, while for the CIR dataset only latter of the two, as during the previous processing the RGB dataset showed higher number of aligned imagery caused by sufficient image overlap. DSM and orthophotos were exported as .TIFF format files with spatial resolution 5 cm/pix. Img. 9. and Img. 10. are showing the sparse point cloud, dense point cloud and mesh of the both datasets.



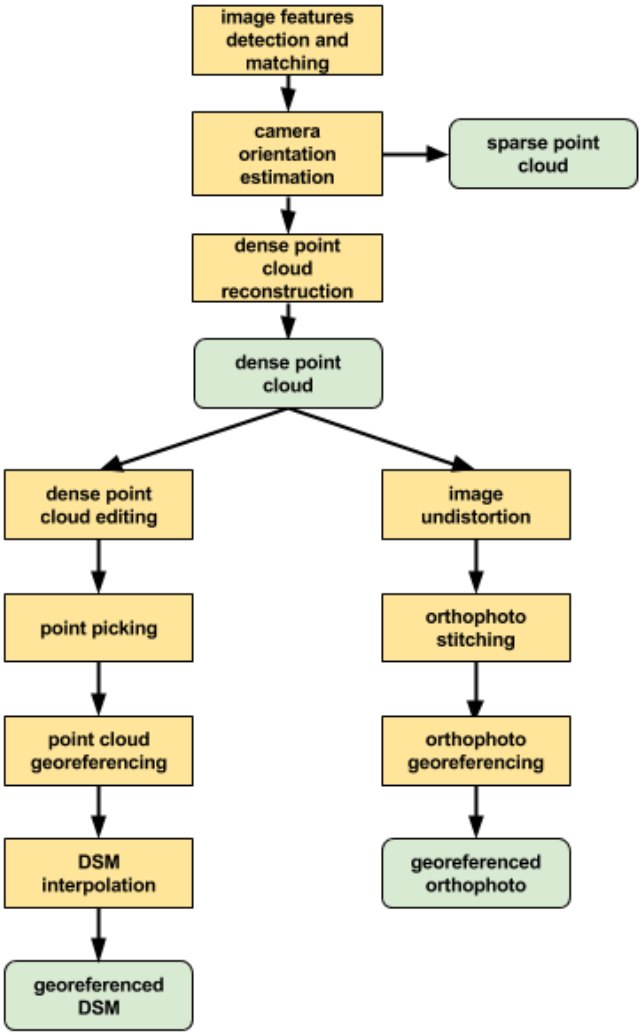
Img. 9.: Sparse point cloud (a), dense point cloud (b) and mesh (c) created based on RGB imagery.



Img. 10.: Sparse point cloud (a), dense point cloud (b) and mesh (c) created based on CIR imagery.

4.3 Image processing using VisualSfM

VisualSfM software presents GUI approach for SfM algorithms execution. In contrast with the previous solution algorithms used in the software are publicly known. General workflow involves image loading followed by the feature detection, matching and bundle adjustment resulting in a sparse point cloud. Once the cloud is generated additional cloud densification can be performed in case the CMVS/PMVS2 software library was previously installed. VisualSfM however provides only limited option for georeferencing, thus in this workflow additional open-source software combination was used for this task. Next, VisualSfM does not provide an option to generate and edit mesh nor export DSM and orthophoto directly. Due to this reasons, the workflow includes usage of CloudCompare (CLOUDCOMPARE, 2015) software in combination with GRASS GIS in order to georeference the point cloud. After the cloud is transformed, interpolation can be performed either using open-source QGIS or commercial ArcGIS software. Orthoimagery generation was achieved using CloudCompare which undistorts input imagery based on camera parameters estimated by VisualSfM. Images processed in this manner can be stitched using free tool MicrosoftICE or similar. Additional georeferencing of the created orthoimagery has to be performed. VisualSfM allows to adjust some of the processing parameters directly from GUI, however full parameter list is stored in *nv.ini* file. Both datasets were processed in the same manner. DSM export was performed only for RGB data due to the same reasons as in the previous workflow. VisualSfM 0.5.26 (64bit) version was used in this workflow. Schema of proposed workflow can be seen on Img. 11.



Img. 11.: Schema of VisualSfM-based workflow.

4.3.1 Feature detection

In order to detect features essential for further camera position estimation VisualSfM takes advantage of the GPU processing-able SiftGPU algorithm. However user can choose to use CPU-based implementations in case of a desire. SiftGPU comes with a wide range of parameters and options, however only a few of them are adjustable by user from the VisualSfM interface. Full list of the algorithms parameters can be found in WU., 2015a.

4.3.1.1 **DIM parameter**

DIM parameter, in VisualSfM *nv.ini* file internally called *maximum_working_dimension* adjusts the dimensions of imagery in case a higher from its two dimensions exceeds given threshold (WU, 2015d). This adjustment is performed due to processing time savings. The parameter can be adjusted manually in case of a higher precision demands and suitable hardware equipment. Dimensioning principle in SiftGPU works as shown below (WU, 2015d):

```
d = max(width, height);
```

```
sz = 2*d
```

$$sz' = \max\{sz / 2^i \mid sz / 2^i <= DIM, int\ i >= 0\}$$

Thus, images are down-sampled or up-sampled prior to the processing dependent on their size. Default value of the DIM parameter is set to *3200*. In such a case, imagery with dimension e. g. 1600 x 800px (d = 1600) would be up-sampled to 3200, while imagery of d = 4000 would be down-sampled to 2000. As the data used in this work are of the dimension 3000x4000px, the *DIM* parameter was set to a value *4096*. Another possibility is to down-sample the data to 3200px however this would lead to a loss of information (WU, 2015d).

4.3.1.2 **TC parameter**

TC parameter represents approximate limit of detectable features per image. It comes with 4 options where (WU, 2015a):

- -TC, -TC1 option keeps the amount of detected features high above the given limit set by numeric value.
- -TC2 is similar to -TC and -TC1, however processing should be performed faster.
- -TC3 keeps the amount of features close to the numeric value.

Setting the value to -1 allows detection without any limitation. Default value of the parameter is set to -TC2 7680, meaning that the amount of detected features should exceed the given value.

The feature detection was performed using CUDA. VisualSfM stores detected features in binary format .SIFT, thus the features can be used or extracted by external program. In this work, TC2 parameter values of the feature detection were set to *4000*, *8000*, *12000* and *-1* (no limit).

4.3.2 Feature matching

Feature matching was performed by the SiftGPU as well. User can choose between CPU or GPU (CUDA or GLSL) implementations. Instead of the full matching, Visual SfM provides option to execute the preemptive version of the matching approach. Additionally software allows to compute matches for a given pairs and to visualize matching matrix expressing intensity of matching relation between all images.

4.3.3 MaxSIFT parameter

Main parameter influencing matching result is the *MaxSIFT*, in *nv.ini* referenced as *param_gpu_match_fmax*. This parameter determines the number of features per image used in matching (WU, 2015d). As the preemptive matching is in use, the features are sorted by scale in a decreasing order. Rising the parameter value thus allows to match also features in a smaller scales. Default value of this parameter is set to 8192, however according to the authors recommendation value 4000 would be sufficient in most of the cases (WU, 2015d).

As in this work the aim was to obtain possibly the best precision results, the parameter was tested using default value 8192 and doubled value of 16384. Matching was performed using CUDA.

4.3.4 Sparse cloud reconstruction

Consequent step after the matching is the bundle adjustment resulting in camera positions estimation and sparse cloud reconstruction. In this step VisualSfM allows to adjust a set of parameters related to the number of iterations in bundle adjustment, frequency of global or partial adjustment execution, MSE threshold or number of imagery used in every partial bundle iteration (WU, 2015d). Sparse cloud reconstruction for every parameter combination described above was performed without adjustment of the BBA parameter. Results of the feature detection, matching and sparse cloud reconstruction are shown in Tab. 4. and Tab. 5.

-TC2 / MaxSIFT	feature detection		feature matching		BBA		
	avg. features detected / img	time [s]	matches	time [s]	sparse cloud points	estimated camera positions	time [s]
4000 / 8192	4765	257	903794	1453	72700	191	74
8000 / 8192	9685	315	847046	3608	113450	195	114
12000 / 8192	14745	285	812779	3884	114310	195	125
no limit / 8192	17764	291	211041	3958	115327	195	122
4000 / 16384	4765	257	523396	1685	73463	192	81
8000 / 16384	9685	315	20869	190	1768	14	4
12000 / 16384	14745	285	102152	3722	11949	76	37
no limit / 16384	17764	291	62973	2077	3064	42	23

Tab. 4.: Results of RGB imagery stitching using different variations of parameters in VisualSfM.

-TC2 / MaxSIFT	feature detection		feature matching		BBA		
	avg. features detected / img	time [s]	matches	time [s]	sparse cloud points	estimated camera positions	time [s]
4000 / 8192	4933	126	100294	802	8045	27	41
8000 / 8192	10440	148	208896	1681	9666	27	48

Tab. 5.: Results of CIR imagery stitching using different variations of parameters in VisualSfM.

-TC2 / MaxSIFT	feature detection		feature matching		BBA		
	avg. features detected / img	time [s]	matches	time [s]	sparse cloud points	estimated camera positions	time [s]
8000 / 8192	10440	148	208896	1681	9666	27	48
12000 / 8192	14571	179	208353	1618	11522	29	41
no limit / 8192	17123	179	208256	1640	13197	35	46
4000 / 16384	4933	126	100294	810	3263	18	40
8000 / 16384	10440	148	36181	851	725	4	4
12000 / 16384	14571	179	21924	983	1029	3	2
no limit / 16384	17123	179	14322	1103	1411	3	2

Tab. 5. - continue: Results of CIR imagery stitching using different variations of parameters in VisualSfM.

Based on the Tab. 4. and Tab. 5. results with the highest number of camera parameter estimations were taken for further processing. In case of the both datasets this was achieved by using -*TC2 / MaxSIFT* parameters setup values *no limit / 8192*.

Prior to the cloud densification, tools for manual alignment of the cameras without estimated parameters were used. VisualSfM provides *Spanning Forest* tool allowing to visualize the matching relationships between imagery. According to the settings, the software can search either for only one model or for several models during the sparse cloud reconstruction. If only one model is searched the result with the highest amount of solved cameras is provided. In case of the multiple search all models are returned, however disconnected. *Spanning tree* tool allows to detect visually the images between models with a weak links. After the pairs of images with the weak linkage are detected, additional feature detection and matching for the given pairs can be executed. In this manner CIR model was manually edited resulting in a merge of two models providing totally 46 solved cameras.

Image pairs with a high amount of a feature matches can be detected in the same way and consequently set as an initialization pair for bundle adjustment, as VisualSfM allows this option.

4.3.5 Cloud densification

Cloud densification was performed using the CMVS/PMVS2 package. In VisualSfM both softwares are used in one step, however the process can be divided into two parts. First CMVS decides the amount of clusters in which the input data are divided. Those are then passed to the PMVS2 for cloud densification. Output is then stored in .PLY files, which amount depends on the amount of the clusters. Densification of RGB dataset results can be seen in Tab. 6.

dataset	num. of clusters	total num. of points	processing time [s]
RGB	5	3904742	10148
CIR	1	1378815	6119

Tab. 6.: Details of point cloud densification using CMVS/PMVS2.

4.3.6 Point cloud georeferencing and DSM export

As VisualSfM does not support direct export of the DSM nor orthoimagery, both products can be obtained using third party software. DSM export was performed only for point cloud derived from the RGB dataset, as more cameras were solved and thus the point cloud provides better area coverage.

Interpolation of the DSM can be performed based on the point cloud, however it has to be georeferenced prior to this step. In order to avoid processing of unnecessary areas in the point cloud and thus achieve processing time savings, the point cloud was first imported to the CloudCompare software where clusters of points resulted from previous steps (.PLY files) were merged together. Areas exceeding area of the interest were clipped. Next, in order to georeference the cloud, local coordinates of GCP points had to be extracted. CloudCompare provides a *Point Picking* tool allowing to achieve the goal. After sufficient amount of point coordinates was captured, GCP file was composed according to the format specifications of V.PLY.RECTIFY module (GRASSGIS, 2015) of GRASS GIS software. This module allows to load, transform and export point clouds in .PLY format. The tool was used iteratively firstly to determine and eliminate RMSE of the transformation by removing GCPs with highest RMSE and then to perform the transformation itself.

Transformed point cloud was imported into ArcGIS software, where *ASCII 3D to Feature Class* tool was used to export the point cloud into GIS format (.SHP file). The same can be achieved using *Add Delimited Text Layer* tool in QGIS. After the import, *Inverse Distance Weighted* (IDW) interpolation was used for the DSM interpolation.

4.3.7 Orthophoto export and georeferencing

Due to the lack of orthophoto export feature in VisualSfM, additional software pipeline had to be utilized. The CloudCompare software allows to import the output file of VisualSfM containing camera parameters estimation (.OUT format). As a part of the import, CloudCompare provides option to undistort imagery based on the parameters contained in the file. Imagery of the both datasets was undistorted in this manner. Consequently, image stitching software Microsoft ICE was used in order to stitch the images together.

This approach requires additional georeferencing, as the image output from the CloudCompare is not georeferenced. The georeferencing can be performed in GIS software such as ArcGIS or QGIS.

5 Acacia tree classification

In this chapter the practical usage of extracted DSM and orthophoto will be demonstrated on the Acacia detection. Products will be enhanced and different classification workflows will be proposed using the RGB and CIR datasets respectively. Classification based on RGB dataset will use the DSM elevation data, while classification using CIR data will use *BNDVI*. Both classifications will be tested with and without inclusion of the IHS bands in order to examine its contribution to the classification accuracy. Two areas will be chosen for classification model development and testing (Segment 1 and Segment 2 respectively). Workflows will use data derived from PhotoScan-based workflow. Classification rulesets were developed using Trimble eCognition Developer 8 software.

5.1 Data enhancements

In order to get advantage of the information provided by *NIR* band in the CIR dataset, *NDVI* based on *Blue* band (referenced as *BNDVI*) was calculated following the formula (1). *Blue* band was chosen due to the lack of a *Red* band in the imagery. Another enhancement was done by transforming both datasets to the IHS color space. Software used for performing these steps was PCI Geomatica (*Raster Calculator* and *RGB to IHS* tools).

Next, absolute height data extracted from the RGB dataset had to be transformed into relative information. Absolute information provided by DSM would produce false assumptions about vegetation height, as the terrain slope would influence final vegetation height. Transformation can be done by subtracting absolute heights of terrain (DTM) from absolute heights of the surface model (DSM) resulting in the *Surface* band. The DTM was generated using PCI Geomatica's *DSM2DTM* algorithm. Raster-to-raster calculation was performed using *Raster Calculator* tool in ArcGIS.

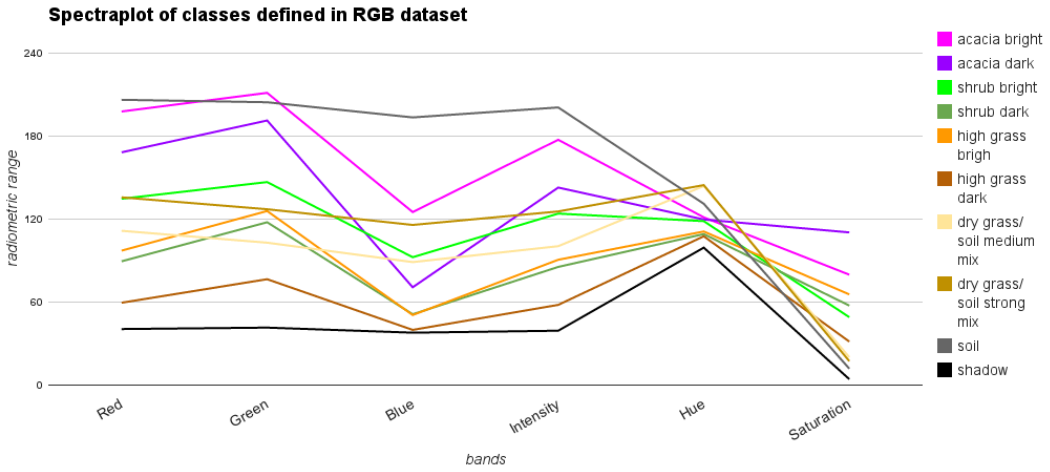
Img. 12.: Diagram of Surface band extraction workflow.

5.2 Classification schema and spectral behavior

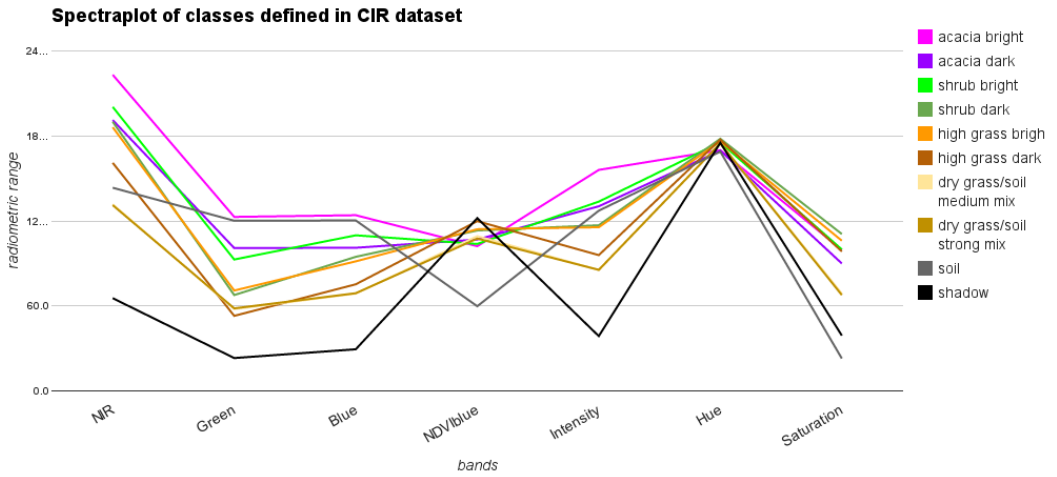
Classification schema was defined based on a main land cover types and their illumination properties observable on the imagery. The classification followed hierarchical structure thus the classes for the both classifications were organized in hierarchies. Each hierarchy consists of two levels: *Classification Level 1* and *Classification Level 2*. Class definitions differ for each dataset.

Classification Level 1 of CIR dataset is formed by two main land cover types present on the imagery - *Non-vegetation* and *Vegetation* superclasses. The former was not examined more in detail, however classes of *Soil* and *Shadows* can be implicitly included in this superclass. *Vegetation* superclass consists of following classes: *Acacia bright, Acacia dark, Shrub bright, Shrub dark, High grass bright, High grass dark, Dry grass/soil mix medium, Dry grass/soil mix strong*.

Classification schema for RGB dataset differ in structure as the classification is based on surface elevation (*Surface* band). *Classification Level 1* is formed by *Low vegetation* and *High vegetation* superclasses. While the former is not a subject of the interest, latter one is subdivided into *Acacia bright, Acacia dark, Shrub bright* and *Shrub dark* children classes. Finally in both workflows classes were aggregated into *Acacia* and *Non-Acacia* classes. The classification and aggregation schema can be seen in Tab. 11. and Tab. 12., Appendix B. In order to get insight into spectral behavior of different classes, spectra plots for both datasets were created (Img. 13 and Img. 14.).



Img. 13.: Spectraplot of classes defined in RGB dataset.



Img. 14.: Spectraplot of classes defined in CIR dataset.

5.3 Classification using RGB data

Classification model based on RGB imagery relies on the surface elevation information extracted. Firstly coarse image segmentation was performed in order to separate *Classification Level 1* classes. Next thresholding based on the *Surface* band was executed. Areas below given threshold were classified as *Low vegetation* while areas above the threshold as *High vegetation* superclass. Fine classification was performed in order to distinguish between vegetation species within *High vegetation* superclass - *Acacia bright, Acacia dark, Shrub bright* and *Shrub dark*. Fine segmentation was applied followed by Nearest Neighbor classification. There were two variants of classification inputs - segments were classified using *Red, Green* and *Blue* band (NN1) or *Red, Green, Blue, Intensity, Hue* and *Saturation* bands (NN2). Classification resulted in 4 outputs: $S1NN1_{RGB}$, $S1NN2_{RGB}$, $S2NN1_{RGB}$ and $S2NN2_{RGB}$. Finally classification results were aggregated according to the aggregation schema shown on Tab. 11., Appendix B. Both segments were classified using same elevation threshold and parameters of OBIA. The proposed workflow is represented on Img. 32., Appendix A.

5.3.1 Coarse classification

In order to obtain image objects of a basic land cover types coarse (without a need to delineate detailed information) segmentation was performed. Parameters of segmentation were estimated based on a visual assessment trying to avoid oversegmentation:

- *Band weights* were set to 1 for all of the bands expect *Hue* band where spectrogram showed small spectral distances between the classes.
- *Scale* parameter was set to 15 in order to obtain objects of larger size.
- *Shape* parameter was set to 0.3 as there is no expectation of regular shapes in the objects.
- *Compactness* was set to 0.7 as a spectral homogeneity is required.

Class assignment based on an elevation thresholding was applied. The value of a threshold was set to 0.7 assigning all objects with the mean elevation lower than 0.7 m to the *Low vegetation* superclass. Consequently inverse assignment was used to classify all unclassified objects as a *High vegetation* class.

5.3.2 Fine classification

Classification distinguishing detailed vegetation types requires segmentation of a lower scale providing objects with high spectral separability. Particularly the aim was to distinguish between *Acacia* and *Shrub* classes assuming these two classes are forming the *High vegetation* superclass. Segmentation was performed over the objects of a H*igh vegetation* class. Parameters of fine segmentation were set as follows:

- *Band weight* of *Hue* was set to 0.2 as this bands provides small class separability which can be observed on the spectrogram (Img. 14.). *Blue* band weight was set to 0.5 due to the possibility of confusion between *Acacia* and *Shrub* classes observable also from the spectrogram (Img. 14.). As the height difference between the *Acacia* and *Shrub* classes is not evident, Surface band weight was set to 0.
- *Scale* parameter was set to 6 in order to obtain objects of small size.
- *Shape* parameter was set to 0.2 as shape is not playing important role in this classification.
- *Compactness* value was determined as 0.8 as a high level of spectral homogeneity is required.

In order to separate *Acacia* and *Shrubs* classes the Nearest Neighbor classification was applied. Prior to this step the class samples were trained.

5.4 Classification using CIR data

The process of the classification using CIR dataset follows hierarchical classification as well, however the classification schema differs from the schema used previously. Firstly, coarse classification was performed dividing image into the objects distinguish between *Vegetation* and *Non-vegetation* classes. Consequently *Non-Vegetation* class was classified by thresholding of *Saturation* band, thus *Soil* and *Shadow* classes were distinguished. Next, using inverse assignment all the objects not being assigned to *Non-vegetation* were assigned to the *Vegetation*. In second step fine classification was performed. Objects of *Vegetation* class were resegmented using fine scale. Samples of the classes were collected and Nearest Neighbor classification was utilized to classify image objects of *Vegetation* superclass in detail. Classification was performed in two variants, where first variant (NN1) input bands were *NIR*, *Green*, *Blue* and *BNDVI* and second variant (NN2) with input bands *NIR*, *Green*, *Blue*, *BNDVI*, *Intensity*, *Hue*, *Saturation*. Results were aggregated following *Aggregation Level 2* scheme (Tab. 12., Appendix B). Process workflow is shown on Img. 33., Appendix A. Same workflow was applied on the both segments resulting in 4 scene classifications respectively: $S1NN1_{CIR}$, $S1NN2_{CIR}$, $S2NN1_{CIR}$ and $S2NN2_{CIR}$.

5.4.1 Coarse classification

Segmentation of the imagery was performed in order to delineate basic land cover types. Particular interest of this segmentation was to identify objects of the *Shadows* and *Soil* classes. Parameters of segmentation were set as follows:

- *Band weights* were set to priority 1 except *Hue* band which priority was lowered to 0.5 due to its lower ability to separate classes.
- *Scale* value was set with respect to the purpose of the coarse classification. The parameter value was set to 15.
- *Shape* was set to 0.3 due to lack of regular shape pattern detectable on the imagery in case of *Soil* and *Shadow* classes.
- As spectral homogeneity is an important requirement, *compactness* parameter was set to high value - 0.7.

After the segmentation, thresholding using *Saturation* band was performed due to the spectral behavior of the classes *Soil* and *Shadows* in this band, where both of them have smallest DN values. The threshold condition was assessed manually and was determined to the value of 55. All objects with mean value of the *Saturation* less or equal to 55 were assigned to *Non-vegetation* class using Assign Class algorithm. Objects not assigned to the *Non-vegetation* yet were assigned to the *Vegetation* using inverse assignment. Algorithm *Assign Class* was utilized again.

5.4.2 Fine Classification

Fine segmentation was executed over objects previously classified as the *Vegetation* class. This objects were segmented into a smaller units by adjusting the scale parameter in the first place. Parameters of the multiresolution segmentation algorithm used in this process were set as follows:

- *Band weights* were set to value 1 in most of the cases. Exception was made in case of *Hue* and *BNDVI* bands which provide small separability between classes, thus their values were set to 0.2 and 0.4 respectively.
- *Scale* parameter was determined to be of significantly smaller number compared to the value used during coarse segmentation in order to obtain small and spectrally homogeneous classes. The value of this parameter was set to 6.
- Due to the small size of the resulting objects there is no expectation of shape pattern to be captured, thus *Shape* parameter was set to 0.2.
- On contrary high compactness level is required in order to separate vegetation species with only gentle differences in spectral behavior. Value of *Compactness* parameter was set to 0.8.

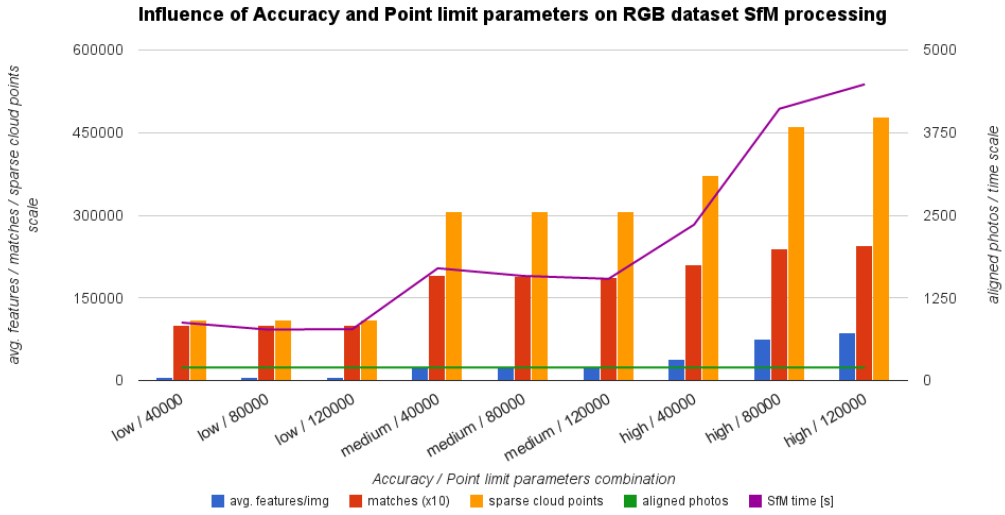
After the fine segmentation the process of classification using Nearest Neighbor algorithm was performed. Prior to this step, samples of the classes were collected. Samples were collected for all of the child classes of the *Vegetation* superclass. Consequently the classification algorithm was executed for the each segment with variable input bands set resulting in $S1NN1_{CIR}$, $S1NN2_{CIR}$, $S2NN1_{CIR}$ and $S2NN2_{CIR}$ classifications.

6 Results and discussion

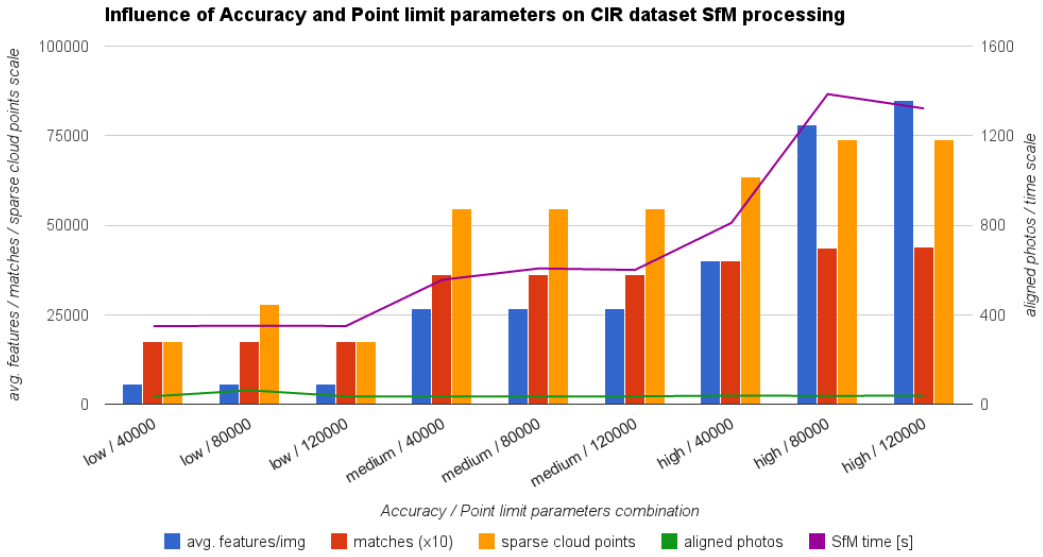
Workflows presented in the previous chapters resulted in extraction of the DSM and orthophoto products. The products were used in order to demonstrate the classification of vegetation. In this chapter, the obtained results will be presented and discussed. Firstly, influence of the parameters of both workflows will be presented and discussed. Next, georeferencing results will be presented. Then orthoimagery resulting from the both workflows will be presented and compared based on a visual assessment. DSMs extracted using both workflows will be compared in terms of a spatial accuracy assessed using GCPs. Comparison between two DSMs will be performed by subtraction. Finally, results of the classification will be presented as well as IHS contribution to the classification accuracy will be examined.

6.1 PhotoScan-based workflow results

Camera parameters estimation was performed using different parameters setup available in the Agisoft PhotoScan software in order to empirically determine the best suitable combination of parameters. While combinations of *Accuracy* and *Point limit* parameters were used, *average feature amount per image*, *number of matches*, *sparse cloud points* amount, *aligned photos* amount and *sfm time* were observed. Same set of the parameters setup was tested on both datasets. The parameters influence was examined in process of the image feature detection, matching and bundle adjustment, as these processes are the most important in SfM workflow and the quality of the consequent steps such as point cloud densification or mesh estimation dependents on the success of these steps. Graphical representation of the obtained results using different parameters setup is presented on Img. 15. and Img. 16.



Img. 15.: Influence of accuracy and point limit parameters on stitching results of CIR dataset.



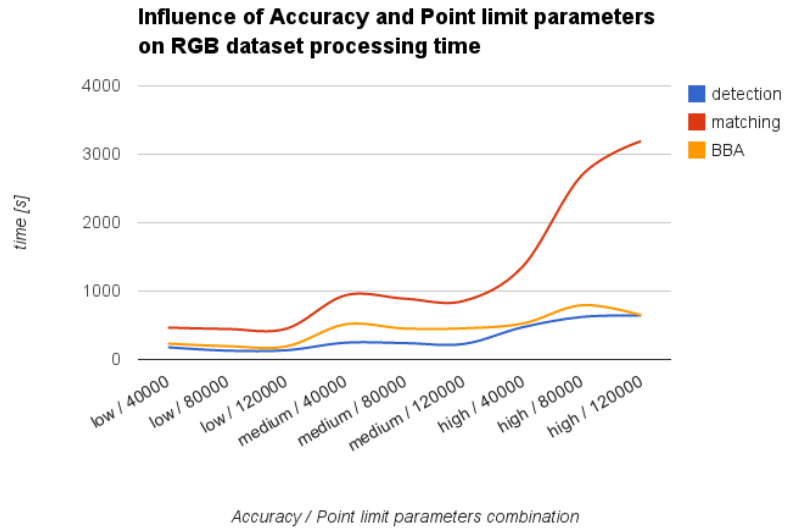
Img. 16.: Influence of accuracy and point limit parameters on stitching results of RGB dataset.

Based on the information provided by the results of stitching tests (Img. 15 and Img. 16.), *Accuracy* parameter has an influence on the *average amount of features / image, matches* number and the *sparse cloud points* amount as well as on the overall processing time. Next, variations in the *Point limit* parameter do not change dependent variables significantly while using constant *Accuracy* parameter. Exception within dependent variables is the *aligned photos* variable, which does not change significantly. Another exception from the previously mentioned conclusion occurs in the case of *Accuracy*: high / *Point limit*: 80000 parameter setup, where number of *tie points*, *sparse cloud points* as well as the *processing time* are rising significantly. Parameter setup *Accuracy*: high / *Point limit*: 120000 does not significantly raise the values of dependent variables parameters. *Overall time* parameter value is, however, still rising.

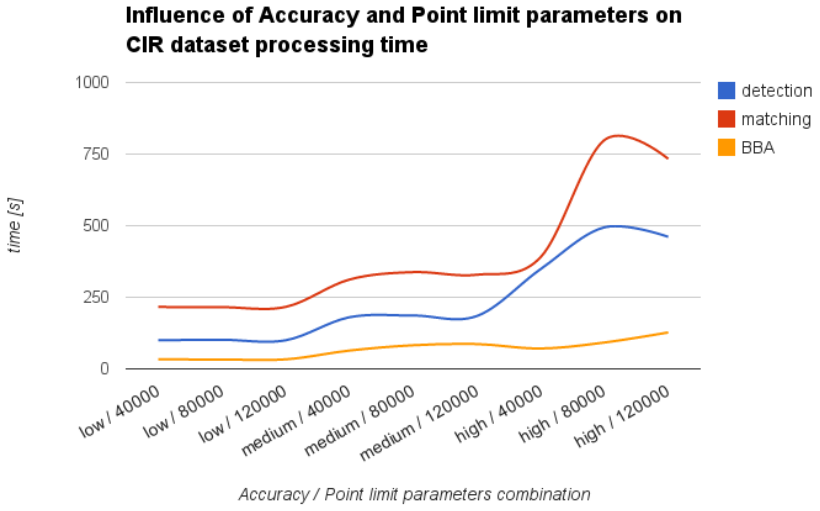
Amount of the *aligned photos* in case of the RGB dataset is constant (195) while interesting behavior in case of the CIR dataset can be observed. Amount of the *aligned photos* in most of the cases is balancing close to the value of 35, while in case of the parameter combinations *Accuracy*: low / *Point limit*: 80000 the amount is nearly doubled - 62 images. This fact can be potentially used in iterative processing of the dataset with lower overlap quality, where initial image estimates can be produced using such a combination of parameters. In next iteration, these estimates can serve as an input for the stitching software in order to maximize amount of the stitched imagery and number of the points detected. Generally amount of the stitched images of the CIR dataset, compared to the RGB dataset is small, which can be caused by small imagery overlap.

Best performance in the case of CIR dataset was observed using combination of parameters *Accuracy*: low / *Point limit*: 80000 with respect to the amount of images with estimated parameters. In case of the RGB dataset, best performance was recorded in the case of parameters combination *Accuracy*: high / *Point limit*: 80000, with respect to the processing time, as the amount of solved cameras is constant. Next, difference in *sparse point cloud* amount, compared to the option using *Accuracy*: high / *Point limit*: 120000 is not significant.

In order to get insight into the relationship between processing time and combination of parameters, the times were recorded for each of the subprocess.



Img. 17.: Processing times of each subprocess in RGB imagery stitching with respect to parameter setup.



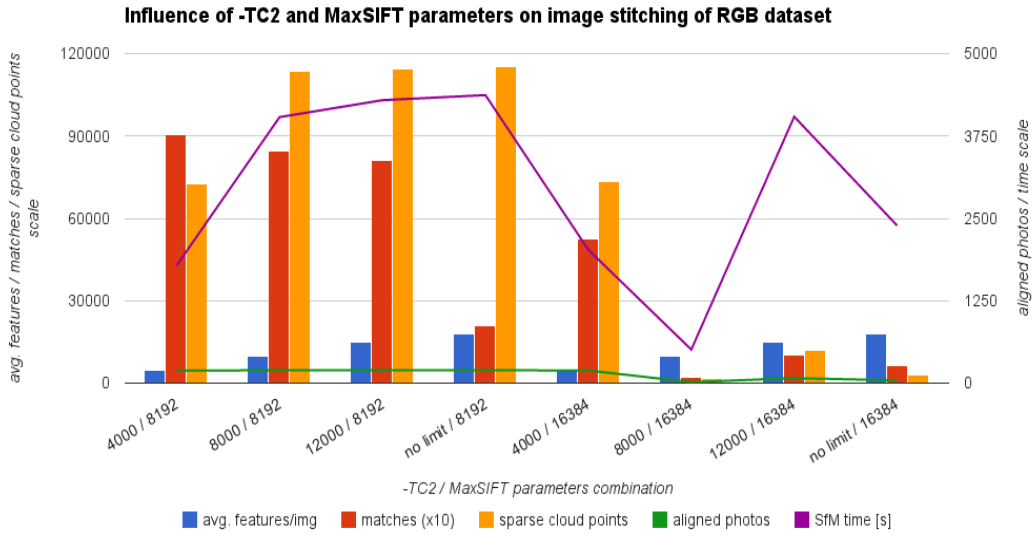
Img. 18.: Processing times of each subprocess in CIR imagery stitching with respect to parameter setup.

Both graphs shown above suggest that the highest processing time requires a step of the feature matching. However, interesting observation occurs in case of the *Point limit* parameter, where its change of the value does not affect significantly the processing times while constant *Accuracy* parameter value is kept. The only exception occurs again in the case of *Accuracy*: high / *Point limit*:

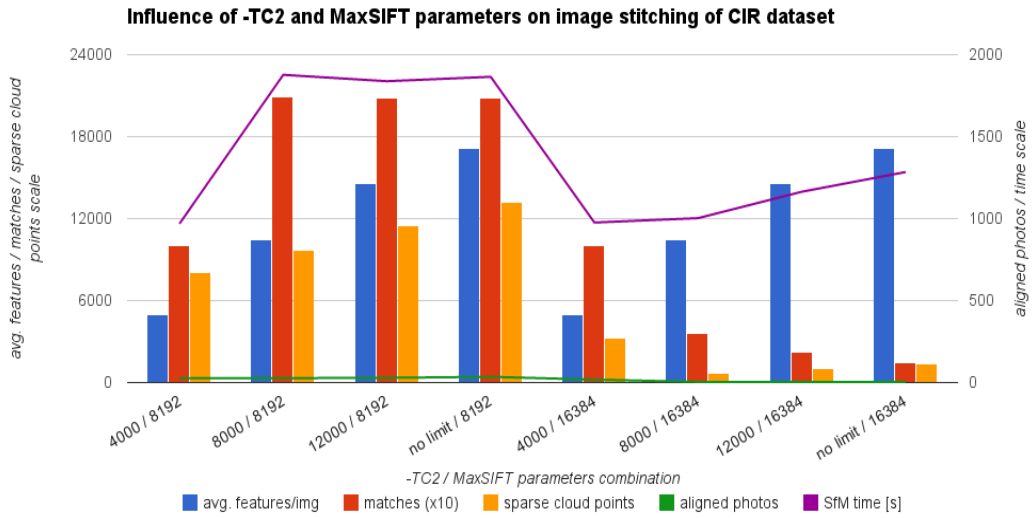
80000, where the processing times start to rise significantly. It can be expected that the feature detection time increase affects the matching time, as the matching algorithm has to process higher amount of the matches. Thus it can be concluded that an average number of features detected per image increases significantly using parameter setup *Accuracy*: high / *Point limit*: 80000 affecting the processing time and amount of sparse cloud points. This conclusion can be used in the case of a high precision requirement, while longer processing times are acceptable and imagery with sufficient overlap is provided.

6.2 VisualSfM-based workflow results

Similarly to the previous case, examination of parameters in case of the workflow based on VisualSfM focused on processes of the feature detection, matching and sparse cloud reconstruction. Values of the parameter *-TC2* influencing the amount of features extracted per image and *MaxSIFT* responsible for amount of the features detected per image were a subject of a variations in the tests. Their relationship towards the same dependent variables as in the previous workflow are shown on Img. 19. and Img. 20.

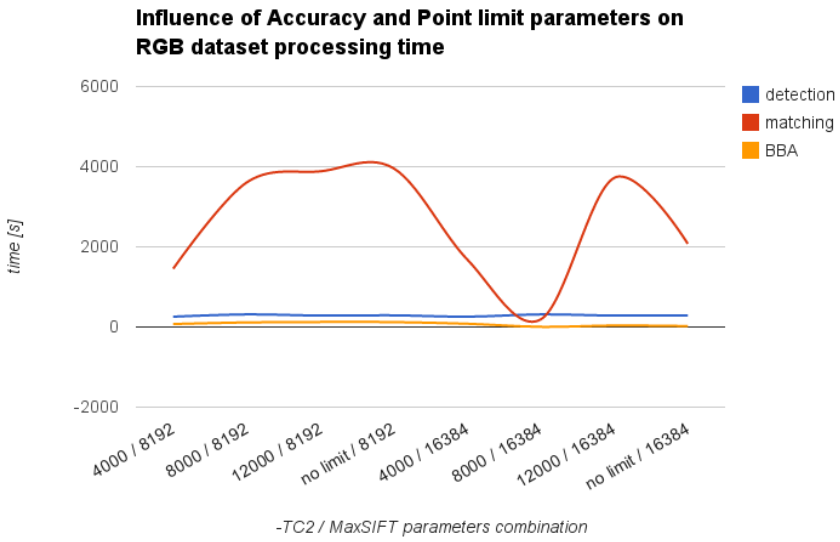


Img. 19.: Influence of -TC2 and MaxSIFT parameters on image stitching in RGB dataset processing.

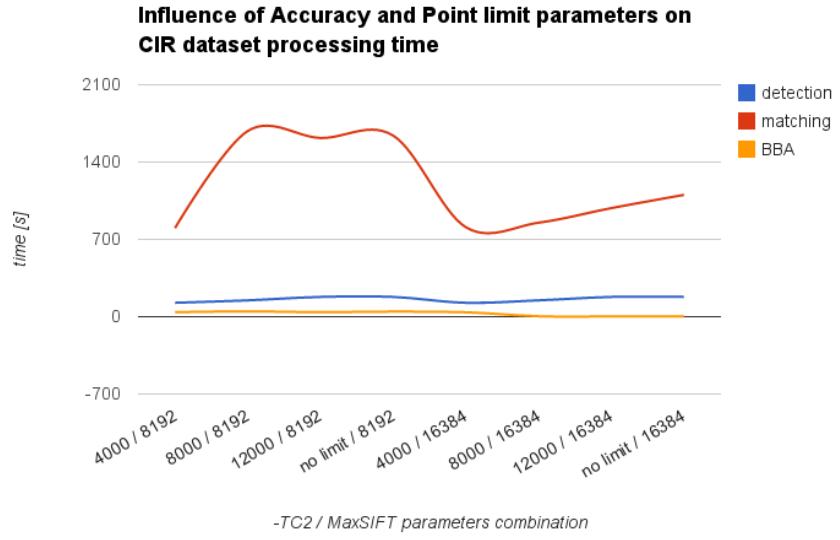


Img. 20.: Influence of -TC2 and MaxSIFT parameters on image stitching in CIR dataset processing.

As expected, the amount of the features detected per image decreases with rising of the *-TC2* parameter value. This trend can be observed without exception in all of the processing tests. As the default value of the *-TC2* parameter in VisualSfM is set to a number close to 8000, it is interesting to observe the amount of features detected when no limitation is provided. Another noticeable observation occurs in the case of the matches amount, where the value is decreasing while *MaxSIFT* parameter is being constant. This trend occurs in the both datasets, except of CIR dataset, *MaxSIFT*: 8192 case. This parameter is responsible for the amount of the features taken into account while matching, thus rise of the matches could be expected with increase of the parameters value. However, results show the opposite situation. Amount of the solved cameras is also higher in case of the *MaxSIFT*: 8192 parameter value, which can be observed in both of the cases. The highest number of solved cameras in case of the RGB dataset occurred in the case of *-TC2/MaxSIFT*: no limit/8192 configuration, however the values did not vary significantly within the group of the tests using *MaxSIFT*: 8192 value. In case of the CIR dataset, mentioned configuration provided the highest amount of solved cameras. Thus, in both of the cases this configuration was used for the further processing. Next, processing times of the separate subprocesses are demonstrated below.



Img. 21.: Processing times of each subprocess in RGB imagery stitching with respect to parameter setup.



Img. 22.: Processing times of each subprocess in CIR imagery stitching with respect to parameter setup.

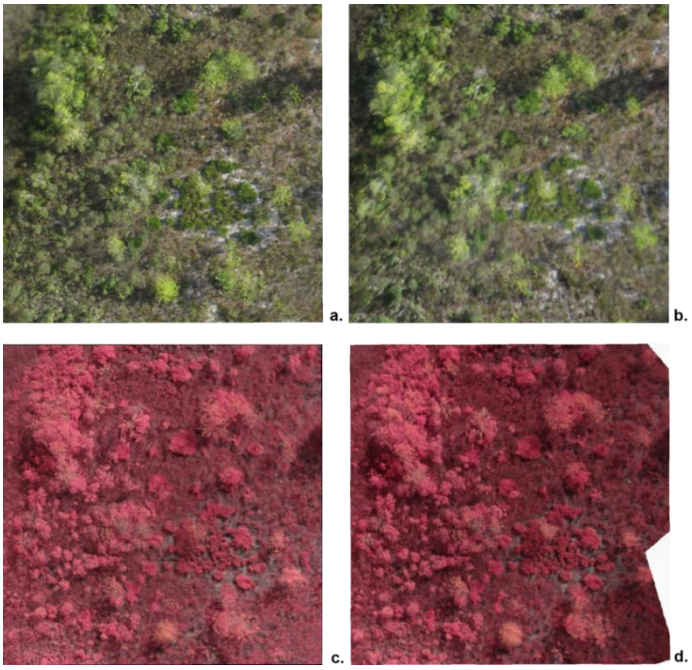
As it can be observed, matching presents the most time-costly subprocess in this part of the workflow. On contrary, detection and BBA processing times are relatively small and their variations are insignificant. Corresponding to the previous parameter behavior, matching time decreases significantly in case of the

configuration -*TC2/MaxSIFT*: 8000/16384. This behavior can be expected as the amount of features matched in this case is small as well.

Concluding the results presented above, it can be said that the value of *MaxSIFT*: 16384 does not provide satisfactory results, however values increase of the dependent variables could be expected. The best solution for processing with higher precision demands appears to be the parameter configuration *no limit/8192*, however this setup requires higher processing times. Keeping the value of *MaxSIFT*: 8192 provides better results in every case, compared to the value of 16384. Thus, in case of a lower precision demands as well as a lower processing time requirement (e. g. data check in field), 4000/8192 configuration can be used.

6.3 Orthophoto results comparison

Both workflows resulted in export of two products - DSM and orthophoto. In this section, orthophotos of the both datasets derived from the workflows will be compared. The comparison will consist of a visual presentation and comparison. Georeferencing of the orthoimagery is well known procedure, thus there will be no attention dedicated to this process. Below are presented the orthophotos capturing area of Segment 1.



Img. 23.: Orthophotos capturing Segment 1 derived from both datasets. a) PhotoScan-based RGB result; b) VisualSfM-based RGB result; c) PhotoScan-based CIR result; d) VisualSfM-based CIR result.

However the VisualSfM-based workflow involves several software procedures, it can be concluded that the product resulting from this workflow was generally of a good quality. Blur and distortion can be observed in some of the areas caused by a stitching using software which expects fixed camera acquisition position. Moreover, VisaulSfM-based workflow provided smaller area covered by the orthophoto in comparison with the PhotoScan-based workflow in case of the CIR imagery. This dataset provides smaller overlap of the raw imagery compared to RGB dataset, thus it can be considered as more difficult to process.

6.4 DSM results comparison

In this section, the DSMs derived from the both workflows will be compared. First, georeferencing results will be presented, then DSMs will be presented. Accuracy assessment was performed with respect to the ground truthing. To demonstrate the difference between both DSMs, VisualSfM-based DSM (DSM_{vsfm}) was subtracted from PhotoScan-based DSM (DSM_{ps}).

6.4.1 Georeferencing precision

While georeferencing the DSM_{vsfm}, CloudCompare software was used in order to identify the GCP position in the point cloud. Compared to this procedure, PhotoScan provides an option to georeference the point cloud using either mesh generated based on the sparse point cloud or directly locating GCPs on the imagery. Thus, the CloudCompare workflow is expected to provide smaller accuracy, as the GCP locations was not clearly visible on the point cloud in all of the cases. Same set of the GCPs was used in both of the cases, however GCPs with higher residuals were removed iteratively until threshold of 1.5 m error was reached. Results of the georeferencing are concluded in Tab. 7. Full list of GCPs used in both workflows is available in Tab. 13. and Tab. 14. in Appendix B.

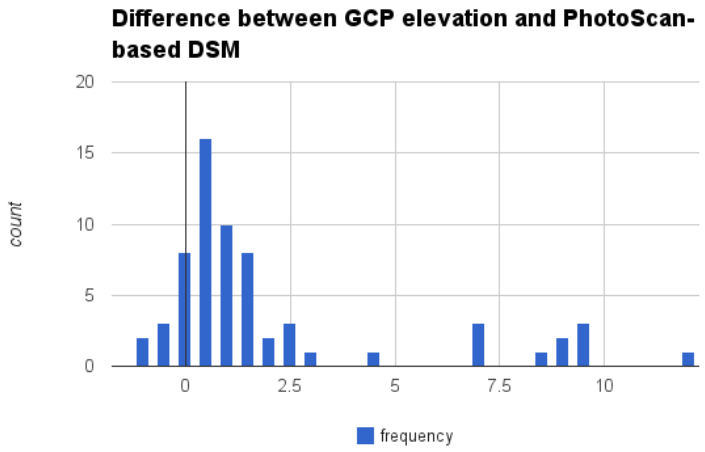
DSM	X error (m)	Y error (m)	Z error (m)	Error (m)
DSM_{ps}	0.3483	0.2914	0.3949	0.6018
DSM_{vsfm}	0.4091	0.4600	0.9478	1.4029

Tab. 7.: X.: X, Y, Z and total error of georeferencing the point clouds.

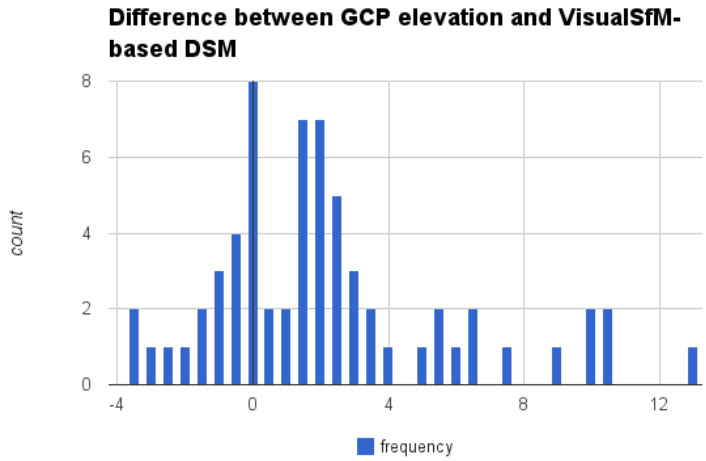
Georeferencing error in all of the directions, as well as total error was smaller in case of PhotoScan-based workflow. The reason expected to be responsible for this result was described in the text above.

6.4.2 Absolute and relative DSM precisions

Absolute elevation precision of the DSMs was assessed by comparison of the DSMs value and subset of a GCPs elevation measurements. A set of 46 GCPs was chosen for the comparison, difference histogram and associated statistics were derived. Img. 24. and Img. 25. are presenting the frequency distribution of the differences. Tab. 8. then shows the statistics of the measurements.



Img. 24.: Elevation difference between GCP and DSMps.



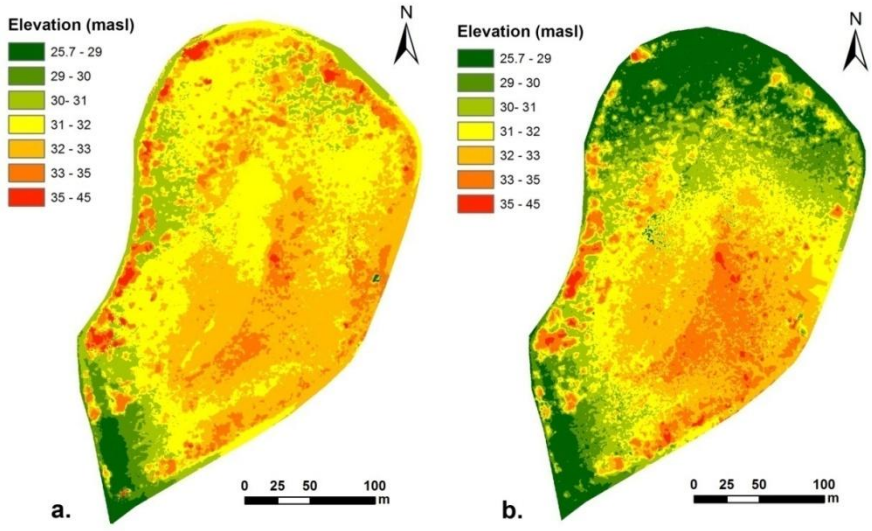
Img. 25.: Elevation difference between GCP and DSMvsfm.

DSM	count	min	max	average	std. dev
GCP - DSM_{ps}	64	-1.27	11.56	1.83	3.09
GCP - DSM_{vsfm}	64	-3.63	12.77	1.97	3.57

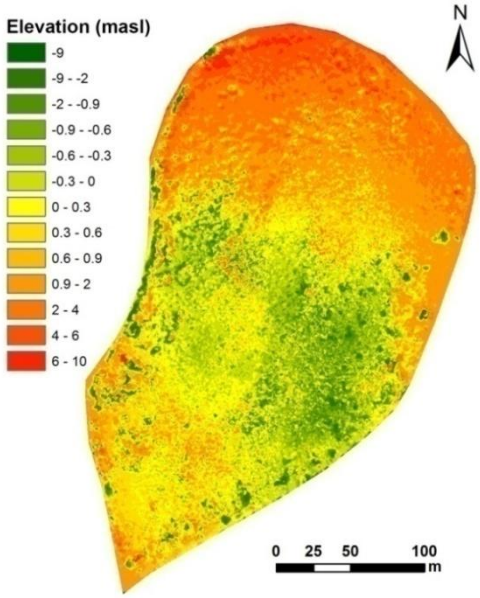
Tab. 8.: Difference statistics of both DSMs.

The highest frequency of the difference values in case of the DSM_{ps} occurs within interval of (0, 0.5) m while the mean of the differences is 1.83 m. As the maximum values reach around 11.5 m, the mean can be significantly influenced by this value. As the histogram is approximately following normal distribution, the values of the differences higher than 5 m can be threaded as an outliers. On contrary histogram of DSM_{vsfm} shows the distribution of a values with two peaks, where 8 cases occur on interval (-0.5, - 0) m and together 14 cases on interval (1, 2) m. Maximum and minimum values of DSM_{vsfm} are larger in respective axes directions, as well as the average, which is higher with 0.14 m compared to the case of DSM_{ps}. Thus lower accuracy of the DSM_{vsfm} in comparison with the DSM_{ps} can be concluded. Average values in both of the cases are positive numbers, thus the DSMs tend to underestimate height of the surface.

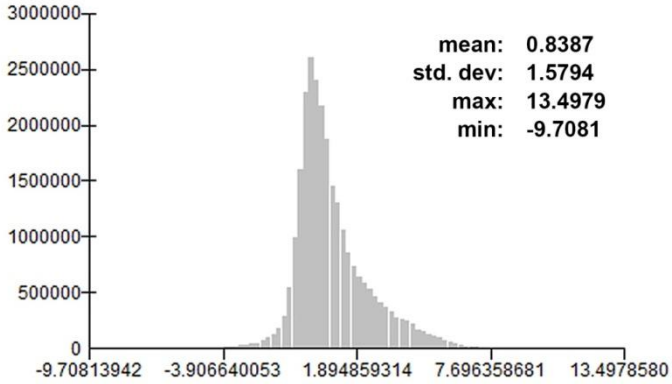
Relative elevation difference between the both DSMs was demonstrated by subtracting the DSM_{vsfm} from DSM_{ps} in order to detect areas with highest differences, as well as areas of identical elevations. Img. 26. shows the resulting DSMs. Img. 27. then shows the difference of the DSMs.



Img. 26.: DSMs created by different workflows; a) DSM_{ps}; b) DSM_{vsfm}



Img. 27.: Difference between DSMps and DSMvsfm.



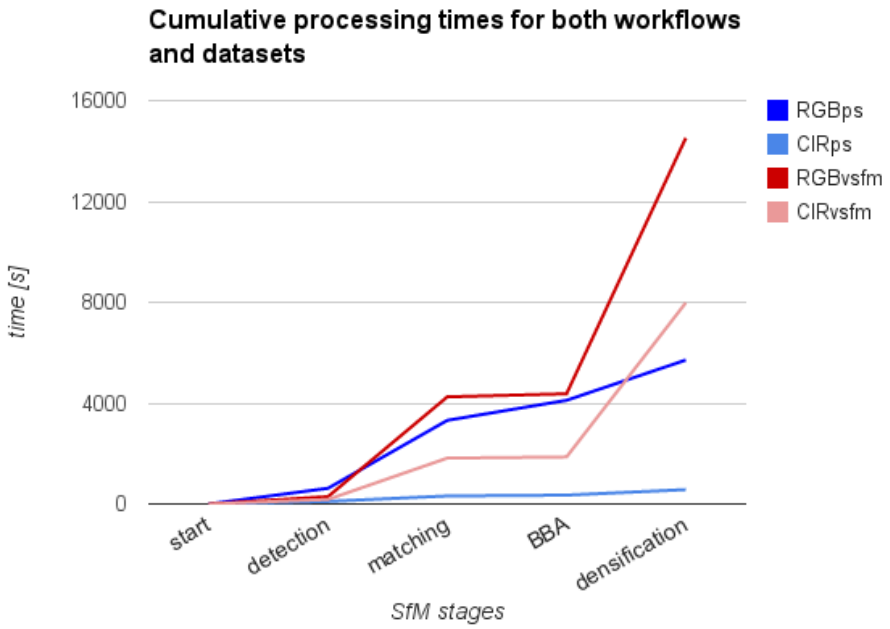
Img. 28.: Statistics of DSM differences.

The areas with the highest differences of the elevation are situated in the northern part of the study area. Central and southern parts of the study area appear to have the highest number of the identical elevation pixels or pixels with absolute difference ranging up to 0.3 m. The average of the elevation difference is 0.84 m and the data follow approximately normal distribution. As the average is positive number, it can be stated that DSM_{ps} generally assigns higher elevations in certain

parts of the study site compared to DSM_{vsfm}. This situation can be caused either by lower precision of latter DSM or by georeferencing error, as there was not significant amount of the GCP distributed in the northern part of the study site. Except the northern part of the area, high differences can be observed also in case of the areas where high elevation is expected, such as the tree peaks. Thus it appears DSM_{vsfm} performs worse in delineating of a fine scale elevation detail as the top elevation areas are smoothed.

6.5 Workflows comparison

Two workflows were presented in the previous text as well as precisions of their results. In this section, workflows processing times will be compared as well as the summary of the discovered facts will be provided. Processing times were recorder for the most suitable parameter setups described in the previous sections. Times were recorded for processes of the SfM until dense cloud reconstruction step. In order to get insight into the processing time costs, Img. 29. presents processing times for the both datasets and workflows.



Img. 29.: Cumulative processing times for both datasets and workflows.

In the both workflows the feature detection and the bundle adjustment represent the least time-expensive operations, while matching and densification are the costly ones. Densification rises the overall processing time significantly in case of the VisaulSfM-based workflow, and thus it is responsible for the significantly higher overall processing time of this workflow.

dataset	workflow	aligned images	dense cloud points	overall time
RGB	Agisoft	195	8279246	5710
	VSFM	190	3904742	14519
CIR	Agisoft	62	7170797	569
	VSFM	46	1378815	7984

Tab. 9.: Summary of obtained results characteristics using different workflows.

Summarizing the obtained results (Tab. 9.), it can be concluded that the both workflows were leading to a successful export of the DSM and orthophoto products. In case of the VisualSfM-based workflow amounts of the stitched images of the both datasets in comparison with PhotoScan-based workflow were smaller. Same can be concluded for the density of the dense point cloud. Processing times were higher in case of VisualSfM-based workflow where main contribution to the processing time was caused by the cloud densification.

PhotoScan-based workflow provides processing of the raw imagery leading to the DSM and orthophoto extraction. In case of the VisualSfM, the pipeline involves several different softwares, thus the overall processing time is significantly higher. However, main advantage of this workflow is the availability of the software without a charge while being used only for a personal or scientific purposes.

6.6 Classification results

Classification workflows resulted in 8 different classifications - two variants of the classification inputs for each dataset type and each segment. In this section the classification results will be assessed in terms of the accuracy using confusion matrices. Variants within the dataset types and within the segments will be compared between each other (e.g. $S1NN1_{CIR}$ vs. $S1NN2_{CIR}$) providing insight into IHS contribution to the classification accuracy.

The assessment was performed over *Aggregation Level 2*. There were 90 samples created manually in order to perform the accuracy assessment. Samples were created and distributed following stratified random sampling strategy (JENSEN, 2005). Minimum samples amount was determined to be placed in the area of each class. This approach is not proportional to the class area percentage (JENSEN, 2005), which is desired as an *Acacia* class covers significantly smaller area compared to a *Non-Acacia* class. Firstly, 30 samples were trained for each of the classes mentioned. In order to capture differences between classification variants (IHS presence and influence on the classification), additional 30 samples were created in a following manner. First, classification results were exported into ArcMap software. Using *Diff* function of *Raster Calculator* tool logically different areas between the rasters were determined on a cell basis (ESRI, 2012). Consequently those were converted into a polygons and their centroids were calculated. Random selection of 30 centroids was included into the sample set of the accuracy assessment.

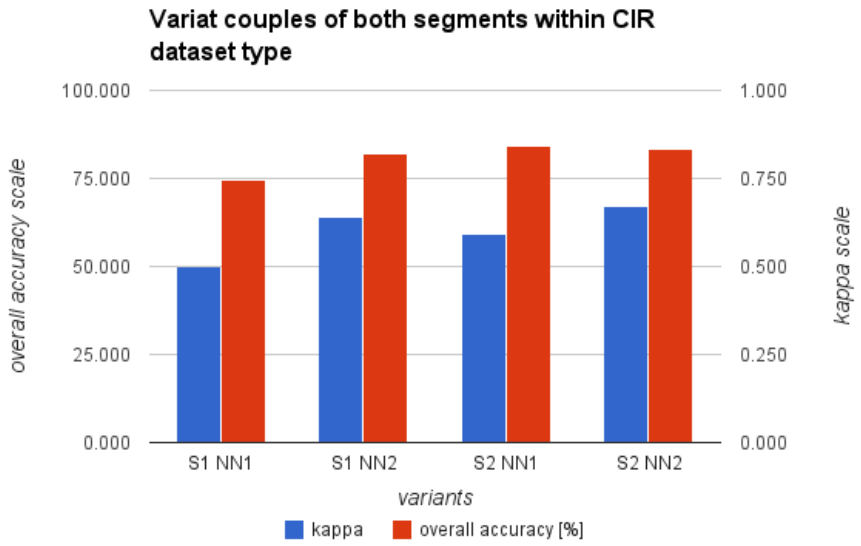
Two characteristics were used in order to measure accuracy. The *overall accuracy* is obtained by dividing the total number of correct pixels by total number of pixels in the matrix (JENSEN, 2005). *Kappa coefficient of agreement* represents an accuracy between the classification map and the reference data. Values of the Kappa coefficient < 0.40 represent poor agreement, values between 0.40 and 0.80 moderate agreement and values over 0.80 represent a strong agreement (JENSEN, 2005).

Confusion matrices are shown in Tab 15. - Tab 22., Appendix B. Resulting accuracies are concluded in Tab. 10. Example of classification can be seen on Img. 34., Appendix A.

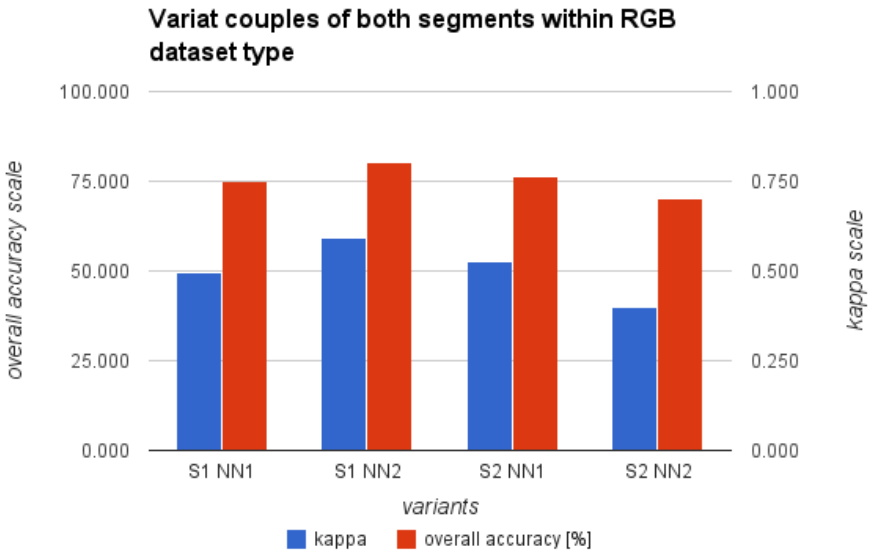
dataset	segment	variant	kappa	overall accuracy [%]
CIR	S1	NN1	0.502	74.444
		NN2	0.640	82.222
	S2	NN1	0.594	84.000
		NN2	0.672	83.333
RGB	S1	NN1	0.496	75.185
		NN2	0.592	80.370
	S2	NN1	0.526	76.296
		NN2	0.398	70.370

Tab. 10.: Classification accuracies for all classification variants.

In order to assess the contribution of the IHS bands as an input to the classification, comparisons of the variant couples within each dataset type and within each segment were performed. As the only difference between variants within each couple is the classification input, the classification accuracy variability is caused by the input bands set.



Img. 30.: Comparisons of the variant couples within CIR dataset type and segment.



Img. 31.: Comparisons of the variant couples within RGB dataset type and segment.

In both segments of the CIR dataset classification variant NN1 returns lower Kappa values compared to variant NN2 (Img. 30.). The maximum reached Kappa occurs in case of the variant $S2NN2_{cir}$, where the coefficients value reaches 0.67. Kappa coefficient in case of the $S1NN2_{cir}$ variant equals 0.64 which is a comparable value. As Kappa coefficients of $S1NN1_{cir}$ and $S2NN1_{cir}$ variants equals to 0.50 and 0.59 respectively, one can state the IHS has a positive influence on the classification accuracy in case of the CIR dataset type.

First pair of the variants in the RGB dataset type, $S1NN1_{rgb}$ and $S1NN2_{rgb}$, is behaving in the similar manner as the correspondent variants in case of the CIR dataset, whereas second couple, $S2NN1_{rgb}$, $S2NN2_{rgb}$, shows a decrease of the accuracy in terms of the *overall Accuracy* as well as *Kappa coefficient* in case of the $S2NN2_{rgb}$ (Img. 31.). This might suggest unsuitability of IHS bands as an input for classification while using the RGB dataset type, however it is not possible to make a curtain conclusion based on the presented result.

7 Conclusion

Workflows for the raw UAV imagery processing using different software - PhotoScan and VisualSfM were presented in this work. Both workflows were successfully used for the DSM and orthophoto products export. Exported data were used in order to demonstrate their usage in Acacia classification. IHS contribution to the classification accuracy result was examined.

Influence of the workflows adjustable parameters on the result quality and processing time was examined. Best parameter combinations for the both workflows were presented.

The orthophotos were compared based on a visual assessment. Products of the both workflows were comparable in terms of quality, however the VisualSfM-based workflow provided orthophoto with smaller are coverage.

The DSM result of the PhotoScan-based workflow showed higher absolute accuracy with respect to the ground truthing, where its error reached 1.83 m, while in case of the VisualSfM-based DSM the error of 1.97 m was recorded. Relative error between both DSMs reached 0.83 m.

Both workflows were compared in terms of results quality and processing time. The PhotoScan-based workflow provided a higher amount of the solved cameras and dense cloud points while requiring less computational time. Another advantage of the workflow was ability to export the products without a need to involve an additional software.

Resulting data were successfully used in the demonstration of Acacia classification. The maximum reached Kappa coefficient value of the classification (0.67) occurred in case of the CIR dataset while using the IHS bands. IHS transformation showed positive contribution to the classification accuracy thus usage of this data enhancement type can be considered in the further work, especially while working with the CIR imagery.

References

AGISOFT LLC., 2014, *Agisoft PhotoScan Help, Version 1.0.4: General workflow: Aligning Photos*, Agisoft LLC, St. Petersburg, Russia.

AGISOFT LLC., 2015, *Agisoft PhotoScan User Manual Professional Edition, Version 1.1* (URL: http://www.agisoft.com/pdf/photoscan-pro_1_1_en.pdf, cited 14/01/2015).

AGISOFT LLC., 2015, *Tutorial (Intermediate level): Orthophoto and DEM generation with Agisoft PhotoScan Pro 1.0.0* (URL: http://www.agisoft.com/pdf/PS_1.0.0%20-Tutorial%20%28IL%29%20-%20Orthophoto_DEM.pdf, cited 18/01/2015).

ESRI, 2012. *ArcGIS Help 10.1: Diff (Spatial Analyst) (URL:* http://resources.arcgis.com/en/help/main/10.1/index.html#//009z000000m3000000)

BANDARA, K. R. M. U., SAMARAKOON, L., SHRESTHA, R. P., KAMIYA, Y., 2011, Automated Generation of Digital Terrain Model using Point Clouds of Digital Surface Model in Forest Area, *Remote Sensing*, 3, 845-858.

BARAZZETTI, L., REMONDINO, F., SCAIONI, M., 2010, Automation in 3D reconstruction: Results on different kinds of close-range blocks. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38, 55-61.

BRODTKORB, A. R., HAGEN, T. R., SÆTRA, M.L., 2013, Graphics processing unit (GPU) programming strategies and trends in GPU computing. *J. Parallel Distrib. Comput*., 73, 4-13.

COLOMINA, I., MOLINA, P., 2014, Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92, 79 - 97.

CLOUDCOMPARE, 2015, 3D *point cloud and mesh processing software*. (URL: http://www.danielgm.net/cc/)

DOBROVOLNY, P., 1998, *Dálkový průzkum země, digitální zpracování obrazu* (Brno: Masaryk university).

D'OLEIRE-OLTMANNS, S., MARZOLFF, I., DIRK, T., BLASCHKE, T., 2014, Detection of Gully-Affected Areas by Applying Object-Based Image Analysis (OBIA) in the Region of Taroudannt, Morocco, *Remote Sensing*, 6 (9), 8287-8309.

DUNFORD, R., MICHEL, K., GAGNAGE, M., PIEGAY, H., 2009, Potential and constraints of Unmanned Aerial Vehicle technology for the characterization of Mediterranean riparian forest.
Int. J. Remote Sens, 30, 4915-4935.

DUTRA, L. V., MENESES, P. R., PARADELLA, E. R., 1988, Color Enhancement of Remote Sensing Imagery Using IHS Transformations and Decorrelation Stretch Methods. *International Society for Photogrammetry and Remote Sensing*, 27, 312-320.

FURUKAWA, Y., CURLESS, B., SEITZ, S. M., SZELISKI, R., 2010a, Towards Internet-scale multi-view stereo. Computer Vision and Pattern Recognition, 1434 - 1441.

FURUKAWA, Y., PONCE, J., 2010b, Accurate, Dense, and Robust Multi-View Stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32, 1362-1376.

GRASSGIS, 2015, GRASS *GIS 7 Addons Manual pages: v.ply.rectify*. (URL: http://grass.osgeo.org/grass70/manuals/addons/v.ply.rectify.html, cited 10/02/2015)

GREEN, S., BEVAN, A., CHAPLAND, M., 2014, A comparative assessment of structure from motion methods for archaeological research. *Journal of Archaeological Science*, 46, 173-181.

GUPTA, S. G., GHONGE, M. M., JAWANDHIYA, P.M., 2013, Review of Unmanned Aircraft System (UAS). *International Journal of Advanced Research in Computer Engineering & Technology*, 2, 1646-1658.

HODGSON, M. E., JENSEN, J. R., TULLIS, J. A., RIORDAN, K. D., ARCHER, C. M., 2003, Synergistic Use of Lidar and Color Aerial Photography for Mapping Urban Parcel Imperviousness, *Photogrammetric Engineering and Remote Sensing*, 69 (9), 973-980.

HOFMANN, P., 2001, Detecting urban features from IKONOS data using an object-oriented
approach, *In Proceedings of RSPS2001*, 79-91.

HUNT, E. R. Jr., HIVELY, W. D., FUJIKAWA, S. J., LINDEN, D. S., DAUGHTRY, C. S. T., MCCARTY, G. W., 2010, Acquisition of NIR-Green-Blue Digital Photographs from Unmanned Aircraft for Crop Monitoring. *Remote Sensing*, 2, 290 - 305.

IBGE, 2015, *RBMC - Brazilian Network for Continuous Monitoring of the GNSS Systems in real time* (URL: http://www.ibge.gov.br/english/geociencias/geodesia/rbmc/ntrip/, cited 19/01/2015)

JACKSON, R. D., HUETE, A. R., 1991, Interpreting vegetation indices. Prev. Vet. Med., 11, 185-200.

JENSEN, J. R., 2005, *Introductory digital image processing: a remote sensing perspective* (3rd ed.) (Upper Saddle River: Pearson Prentice Hall).

KARCHER, D. E., RICHARDSON, M. D., 2003, Quantifying Turfgrass Color Using Digital Image Analysis. *Crop Science*, 43 (3), 943-951.

KHRONOS, 2015, *The open standard for parallel programming of heterogeneous systems*. (URL: https://www.khronos.org/opencl/, cited: 10/02/2015)

KOUTSIAS, N., KARTERIS, M., CHUVIECO, E., 2000, The Use of Intensity-Hue-Saturation Transformation of Landsat-5 Thematic Mapper Data for Burned Land Mapping, *Photogrammetric Engineering and Remote Sensing*, 66, 829-839.

LALIBERTE, A. S., GOFORTH, M. A., STEELE, C. M., RANGO, A., 2011, Multispectral Remote Sensing from Unmanned Aircraft: Image Processing Workflows and Applications for Rangeland Environments. *Remote Sensing*, 3, 2529-2551.

LALIBERTE, A. S., RANGO, A., HERRICK, J. E., FREDRICKSON, E. L., BURKETT, L., 2007, An object-based image analysis approach for determining fractional cover of senescent and green vegetation with digital plot photography. *Journal of Arid Environments*, 69 (1), 1-14.

LALIBERTE, A. S., RANGO, A., 2008, Incorporation of texture, intensity, hue and saturation for rangeland monitoring with unmanned aerial imagery. *In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Canada: GEOBIA 2008), 4/C1.

LALIBERTE, A. S., RANGO, A., 2009, Texture and Scale in Object-Based Analysis of Subdecimeter Resolution Unmanned Aerial Vehicle (UAV) Imagery, *IEEE Transactions on Geoscience and Remote Sensing*, 47 (3), 761-770.

LALIBERTE, A. S., RANGO, A., 2011, Image Processing and Classification Procedures for Analysis of Sub-decimeter Imagery Acquired with an Unmanned Aircraft over Arid Rangelands. GIScience & Remote Sensing, 48, 4-32.

LEHMANN, J. R. K., NIEBERDING., F., PRINZ, T., KNOTH, C., 2015, Analysis of Unmanned Aerial System-Based CIR Images in Forestry- A New Perspective to Monitor Pest Infestation Levels. *Forests Journal* (accepted).

LILLESAND, T. M., KIEFER, R. W., 1994, *Remote sensing and image interpretation* (New York: John Wiley & Sons).

LOWE, D. G., 1999, Object Recognition from Local Scale-Invariant Features. *International Conference on Computer Vision*, 2, 1150-2257.

NASA, 2015, *Measuring Vegetation (NDVI & EVI): Normalized Difference Vegetation Index (NDVI).*(URL: http://earthobservatory.nasa.gov/Features/MeasuringVegetation/measuring_vegetation_2.php, cited 08/02/2015).

NVIDIA, 2015: What is CUDA?. (URL: http://www.nvidia.com/object/cuda_home_new.html#sthash.XhYqzxKC.dpuf, cited: 10/02/2015)

OPENGL, 2015a, *OpenGL Overview*. (URL: https://www.opengl.org/about/, cited: 10/02/2015)

OPENGL, 2015b, OpenGL Shading Language. (URL: https://www.opengl.org/documentation/glsl/, cited: 10/02/2015)

OWENS, J. D., HOUSTOUN, M., LUEBKE, D., GREEN, S., STONE, J. E., PHILLIPS, J. C., 2008, GPU Computing. *Proceedings of the IEEE*, 96 (5), 879-899.

PCI GEOMATICS, 2014, *Geomatica version 2014 Help: DSM2DTM*, PCI Geomatics, Markham, Ontario, USA

PCI GEOMATICS, 2015, *DEM Extraction: Pleiades DEM extraction and DSM to DTM conversion* (http://www.pcigeomatics.com/resources-support/geomatica/tutorials, cited 15.01.2015).

SAPORETTI-JUNIOR, A. W., SCHAEFER, E. G. R., DE SOUZA, A. L., SOARES, M. P., ARAUJO, D. S. D. et al., 2012, Influence of Soil Physical Properties on Plants of the Mussununga Ecosystem, Brazil, Folia Geobotanica, 47 (1), 29-39.

SCARPINO, M., 2011, *OpenCL in action*. (Shelter Islands, NY: Manning Publications Co.).

SHIBA, M., ITAYA, A., 2006, Using eCognition for improved forest management and monitoring systems in precision forestry. *In: Ackerman, P. A., Langein. D. W., Antonides, M. C.*

SHREINER, D., SELLERS, G., KESSENICH, J., LINCEA-KANE, B., 2013, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3.* (8th ed.). (Upper Saddle River, NJ: Addison-Wesley)

SNAVELY, N., SEITZ, S. M., SZELISKI, R, 2006, Photo Tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics SIGGRAPH '06*, 835-846.

SNAVELY, N., SEITZ, S. M., SZELISKI, R, 2007, Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*. 80, 189-210.

SNAVELY, N., SEITZ, S., M., SZELISKI, R, 2008, Bundler: Structure from Motion (SfM) for Unordered Image Collections (URL: http://www.cs.cornell.edu/~snavely/bundler/#S3, Retrieved 14/01/2015).

TORRES, J. C., ARROYO, G., ROMO, C., DE HARI J., 2012, 3D Digitization using Structure from Motion. CEIG - Spanish Computer Graphics Conference.

TRIMBLE, 2011a, *eCognition Developer 8.64.1 User Guide*, Trimble, Munich, Germany.

TRIMBLE, 2011b, *eCognition Developer 8.64.1 Reference Book*, Trimble, Munich, Germany.

TURNER, D., LUCIEER, A., MALENOVSKY, Z., KING, D., H., 2014, Spatial Co-Registration of Ultra-High Resolution Visible, Multispectral and Thermal Images Acquired with a Micro-UAV over Antarctic Moss Beds. *Remote Sensing*, 6, 4003, 4024.

TURNER, D., LUCIEER, A., WATSON, CH., 2012, An Automated Technique for Generating Georectified Mosaics from Ultra-High Resolution Unmanned Aerial Vehicle (UAV) Imagery, Based on Structure from Motion (SfM) Point Clouds. *Remote Sensing*, 4, 1392-1410.

ULLMAN, S., The Interpretation of Structure from Motion, 1979. In *Proceedings B*, 203, 405-426.

VERHOEVEN, G., 2011, Taking Computer Vision Aloft - Archaeological Three-dimensional Reconstructions from Aerial Photographs with PhotoScan. *Archaeological Prospection*, 18, 67–73.

VERHOEVEN, G., DONEUS, M., BRIESE, CH., VERMEULEN, F., 2012, Mapping by matching: a computer vision-based approach to fast and accurate georeferencing of archaeological aerial photographs. *Journal of Archaeological Science*, 39, 2060-2070.

WATTS, A. C., AMBROSIA, V. G., HINKLEY, E. A., 2012, Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and Considerations of Use. *Remote Sensing*, 4, 1671-1692.

WESTOBY, M. J., BRASINGTON, J., GLASSER, N. F., HAMBREY, M. J., REYNOLDS, J. M., 2012, "Structure-from-Motion" photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179, 300-314.

WU, CH., AGARWAL, S., CURLESS, B., SEITZ, S., M., 2011, Multicore Bundle Adjustment. *International Conference on Computer Vision and Pattern Recognition*, 3057 - 3064.

WU, CH., 2013, Towards Linear-time Incremental Structure From Motion. *International Conference on 3D Vision - 3DV*, 127-134.

WU, CH., 2015a *SiftGPU Manual*. (URL: http://www.cs.unc.edu/~ccwu/siftgpu/manual.pdf, cited 09/02/2015).

WU, CH., 2015b, *SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT)*. (URL: http://www.cs.unc.edu/~ccwu/siftgpu/, cited 09/02/2015).

WU, CH., 2015c, *Multicore Bundle Adjustment Manual*. (URL: http://grail.cs.washington.edu/projects/mcba/manual.pdf, cited 09/02/2015).

WU, CH, 2015d, *VisualSFM Documentation.* (URL: http://ccwu.me/vsfm/doc.html#hardware, cited 10/02/2015)

WU, CH., 2015e, *VisualSfM: Install instructions*. (URL: http://ccwu.me/vsfm/install.html, cited 09/02/2015).

WU, CH., 2015f, *VisualSfM:* A Visual Structure from Motion System. (URL: http://ccwu.me/vsfm/index.html, cited 09/02/2015).

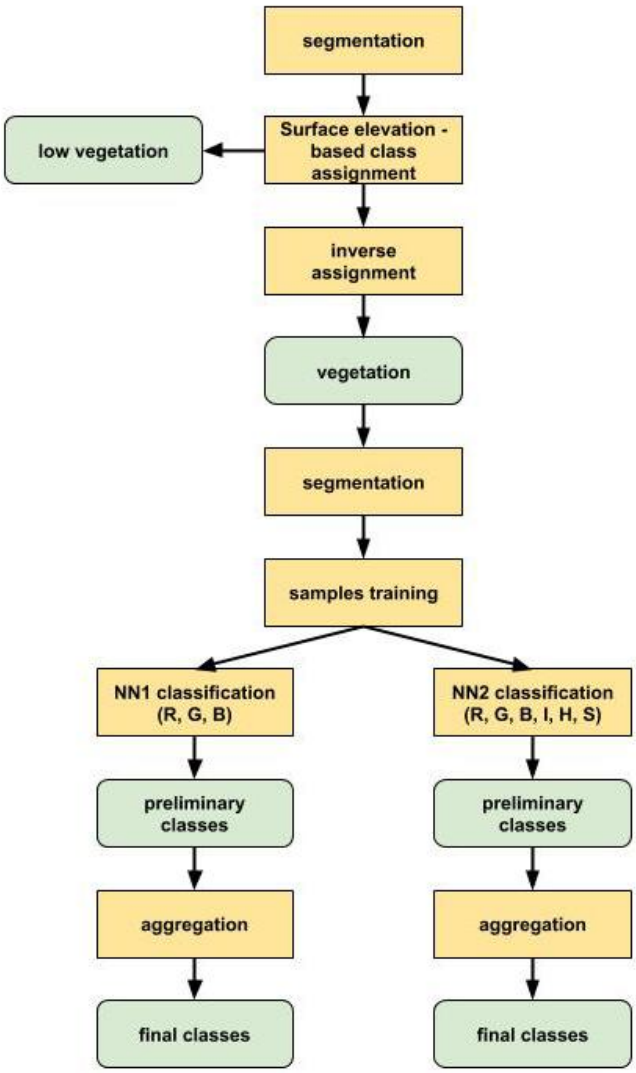
WWU MUENSTER, 2015, *INSPECTED.NET - INvasive SPecies Evaluation & ConTrol NETwork.*(URL: http://www.uni-muenster.de/OekologischePlanung/en/forschung/landschaftsveraenderungen.html, cited: 21/025/2015).

YU, Q., GONG, P., CLINTON, N., BIGING, G., KELLY, M., SCHIROKAUER, D., 2006, Object-based Detailed Vegetation Classification with Airborne High Spatial Resolution Remote Sensing Imagery, *Photogrammetric Engineering and Remote*, 7, 799-811.

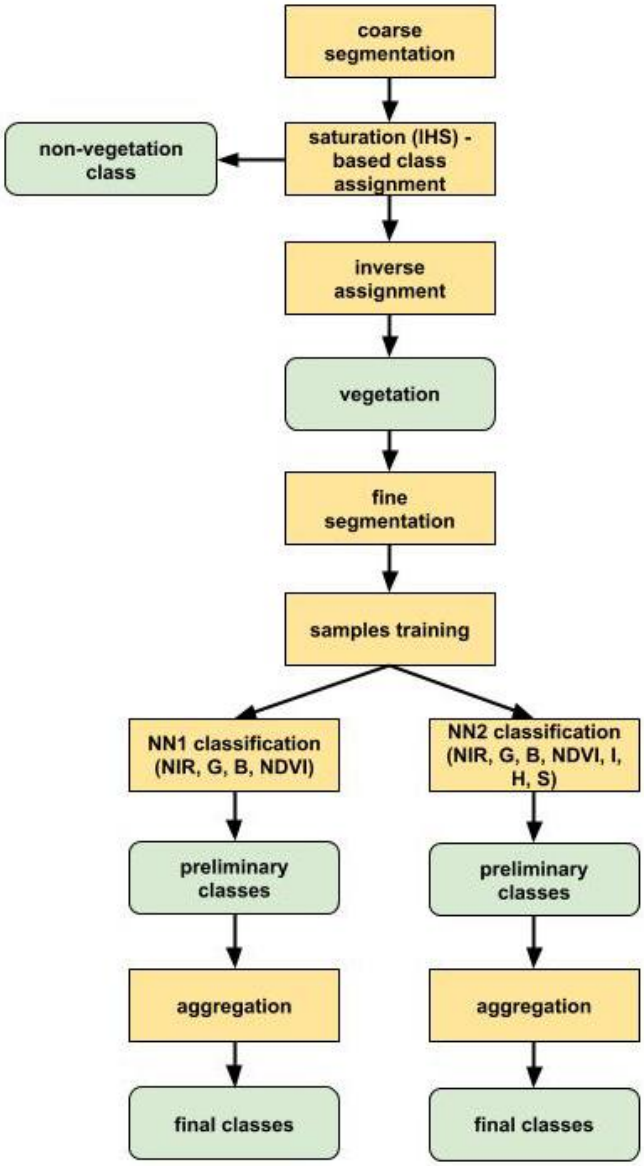
ZHANG, Y., XIONG, J., HAO, L., 2011. Photogrammetric processing of low-altitude images acquired by unpiloted aerial vehicles. *Photogrammetric Record*, 26, 190–211.

Appendix A

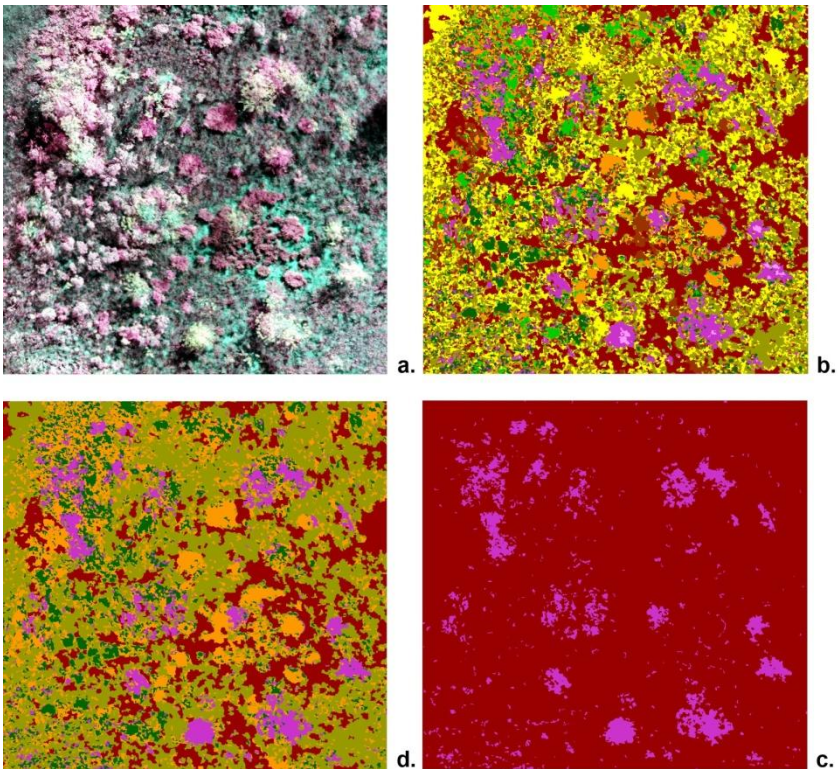
Graphical attachments



Img. 32: Classification workflow for RGB dataset.



Img. 33: Classification workflow for CIR dataset.



Img. 34.: Stages of CIR dataset classification (S1). Original image (a), Classification Level 2 (b), Aggregation Level 1 (c), Aggregation Level 2 (d).

Appendix B

Tabular attachments

RGB dataset							
Classification Level 1		**Classification Level 2**		**Aggregation Level 1**		**Aggregation Level 2**	
name	**color**	**name**	**color**	**name**	**color**	**name**	**color**
high vegetation		acacia bright		acacia		acacia	
		acacia dark					
		shrub bright		shrub		non - acacia	
		shrub dark					
low vegetation		low vegetation		low vegetation			

Tab. 11.: Classification and aggregation schema for RGB dataset.

CIR dataset							
Classification Level 1		**Classification Level 2**		**Aggregation Level 1**		**Aggregation Level 2**	
name	**color**	**name**	**color**	**name**	**color**	**name**	**color**
vegetation		acacia bright		acacia		acacia	
		acacia dark					
		shrub bright		shrub		non - acacia	
		shrub dark					
		high grass brigh		high grass			
		high grass dark					
		dry grass/soil medium mix		dry grass			
		dry grass/soil strong mix					
non - vegetation		non - vegetation		non - vegetation			

Tab. 12.: Classification and aggregation schema for CIR dataset.

Label	X error (m)	Y error (m)	Z error (m)	Error (m)
AC101	-0.037852	-0.181733	1.088911	1.10462
AC230	-0.215477	-0.509633	0.072741	0.558075
AC271	-0.145111	0.196695	-0.072013	0.254818
AC379	0.114029	0.381083	0.198116	0.444383
AC433	-0.014491	-0.103383	0.093471	0.140125
AC053	-0.212997	0.031355	-0.127326	0.250126
AC083	0.375521	0.27513	-0.161709	0.492811
AC099	0.475151	-0.313822	-0.91625	1.07878
AC093	-0.498997	0.144502	0.188562	0.552661
AC103	-0.858558	0.648376	-0.13195	1.083939
AC104	0.219768	-0.047081	0.003172	0.224777
AC145	0.202719	0.046805	0.003075	0.208074
AC353	0.314649	-0.181848	0.045758	0.366288
AC411	0.170711	-0.218275	-0.041838	0.280244
Total	**0.348255**	**0.291414**	**0.394869**	**0.601768**

Tab. 13.: Results of georeferencing in PhotoScan-based workflow.

GCPs				
Label	X error (m)	Y error (m)	Z error (m)	Error (m)
AC053	0.170076	0.185529	0.606472	0.6566
AC083	0.544386	0.438112	0.102369	0.7062
AC093	0.074009	0.255565	0.157015	0.3089
AC099	0.218247	0.483837	1.358026	1.4581
AC101	0.304014	0.906057	0.471677	1.0658
AC145	0.492622	1.191764	1.731138	2.1587
AC230	0.021198	0.423423	1.300352	1.3677
AC271	0.236717	0.111511	2.183428	2.1991
AC353	1.417514	1.373939	0.707312	2.097
AC379	0.549995	0.007298	0.670063	0.8669
AC411	0.782463	0.086563	0.978709	1.256
AC433	0.097803	0.05676	1.106629	1.1124
Total	0.409087	0.46003	0.947766	1.4029

Tab. 14.: Results of georeferencing in VisualSfM-based workflow.

classified	reference data		
	acacia	non - acacia	totals
acacia	32	4	36
non - acacia	19	35	54
unknown	0	0	0
totals	51	39	90

Tab. 15.: Confusion matrix $S1NN1_{cir}$.

classified	reference data		
	acacia	non - acacia	totals
acacia	42	7	49
non - acacia	9	32	41
unknown	0	0	0
totals	51	39	90

Tab. 16.: Confusion matrix $S1NN2_{cir}$.

classified	reference data		
	acacia	non - acacia	totals
acacia	27	15	42
non - acacia	7	41	48
unknown	0	0	0
totals	34	56	79

Tab. 17.: Confusion matrix $S1NN1_{rgb}$.

classified	reference data		
	acacia	non - acacia	totals
acacia	27	11	38
non - acacia	7	45	52
unknown	0	18	0
totals	34	37	90

Tab. 18.: Confusion matrix $S1NN2_{rgb}$.

classified	reference data		
	acacia	non - acacia	totals
acacia	38	7	45
non - acacia	13	32	45
unknown	0	0	0
totals	51	39	90

Tab. 19.: Confusion matrix $S2NN1_{cir}$.

classified	reference data		
	acacia	non - acacia	totals
acacia	38	2	40
non - acacia	13	37	50
unknown	0	0	0
totals	51	39	90

Tab. 20.: Confusion matrix $S2NN2_{cir}$.

classified	reference data		
	acacia	non - acacia	totals
acacia	34	12	46
non - acacia	9	35	44
unknown	0	0	0
totals	43	41	90

Tab. 21.: Confusion matrix $S2NN1_{rgb}$.

classified	reference data		
	acacia	non - acacia	totals
acacia	23	7	30
non - acacia	20	40	60
unknown	0	0	0
totals	43	47	90

Tab. 22.: Confusion matrix $S2NN2_{rgb}$.